

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 РЕФАКТОРИНГ ПРОГРАМНО-ІНФОРМАЦІЙНОГО ПРОДУКТУ
ЗАСОБАМИ PYTHON

Виконав:

ст. гр. IT-32

Шоха А.А.

Прийняв:

Щербак С.С.

Львів-2023

Мета роботи: розробка програмно-інформаційного продукту засобами Python.

Завдання на лабораторну роботу

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 репо

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

Хід роботи

Створений runner для запуску всіх лабораторних:

```
from ui.menu.menu_facade import MenuFacade
if __name__ == '__main__':
    menuFacade = MenuFacade()
    menuFacade.start()
```

Створений фасад для всіх меню:

```
"""
The module provides a text-based interactive menu for navigating through HTML
documentation.
"""
import os
import sys
```

```

sys.path.append('/Users/admin/Desktop/lpnu/5 cem/Specialised programming
languages/source')
from config.paths_config import HTML_DATA
from ui.menu.docs_menu import DocsMenu
from ui.menu.lab2.calculator_menu import CalculatorMenu
from ui.menu.lab3.ascii_generator_menu import AsciiArtMenu
from ui.menu.lab5.cube_menu import CubeMenu
from ui.menu.lab7.google_books_api_menu import GoogleBooksApiMenu
from ui.menu.lab8.plot_menu import PlotMenu
class MenuFacade:
    """
    Facade class for managing a collection of menus.
    This class provides a simple user interface to interact with a collection
of menus.
    Menus are represented as tuples of menu names and menu instances. The
class allows
    the user to select and execute a menu based on their choice.
    """
    def __init__(self):
        """
        Initialize the MenuFacade instance with a list of menus.
        """
        self.__menus = [("Calculator", CalculatorMenu()),
                        ("Ascii Art Generator", AsciiArtMenu()),
                        ("CubeMenu", CubeMenu()),
                        ("PlotMenu", PlotMenu()),
                        ("Google Books API Menu", GoogleBooksApiMenu())]

        self.__finish_number = 0
    def print_menu_options(self):
        """
        Print the available menu options to the console.
        This method iterates through the list of menus and prints each menu
option
        along with its corresponding number.
        Returns:
            None
        """
        for index, (name, _) in enumerate(self.__menus, start=1):
            print(f"{index}. {name}")
        print(f"{self.__finish_number}. Exit")
    def start(self):
        """
        Start the menu façade, allowing the user to interact with the menus.
        This method enters a loop where it continuously prints the menu
options and
        prompts the user for their choice. It then executes the selected
menu.
        Returns:
            None
        """
        while True:
            self.print_menu_options()

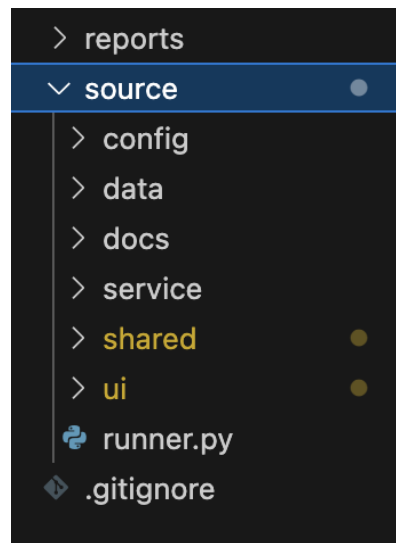
```

```

choice = input("Enter your choice: ")
try:
    choice = int(choice)
    if choice == self.__finish_number:
        break
    if not 1 <= choice <= len(self.__menus):
        raise ValueError
    _, menu = self.__menus[choice - 1]
    menu.run()
except ValueError:
    print("Invalid choice. Enter again!")

```

Оновлена структура продукту:



Висновок: під час виконання програмно-інформаційного продукту засобами Python. лабораторної роботи було розроблено