Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №7

З дисципліни «**Технології розроблення програмного забезпечення**»

Тема: «**ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE METHOD»**»

Варіант №6

Виконав
студент групи ІА–13:
Костенко П.С.

Перевірив:
Мягкий М. Ю.

Київ 2023

Тема:

..6 Web-browser (proxy, chain of responsibility, factory method, template method, visitor, p2p)

Веб-браузер повинен мати можливість зробити наступне: мати адресний рядок для введення адреси сайту, переміщатися і відображати структуру html документа, переглядати підключений javascript та css файли, перегляд всіх підключених ресурсів (зображень), коректна обробка відповідей з сервера (коди відповідей HTTP) - переходи при перенаправленнях, відображення сторінок 404 і 502/503.

Завдання:

1. Реалізувати не менше 3х класів згідно з вибраною темою.
2. Реалізувати один з розглянутих шаблонів по вибраній темі.

Хід роботи:

За моїм варіантом необхідно реалізувати template method. Суть цього шаблону полягає в тому, що при наявності певного алгоритму роботи можливі окремі реалізації його частин у різних ситуаціях, що і передбачається шаблоном.

Було створено абстрактний клас PageCompiler:

```
from abc import ABC, abstractmethod
8 usages
class PageCompiler(ABC):
    1 usage
    def header_compile(self):
        pass
    1 usage
    def footer_compile(self):
        pass
    1 usage
    @abstractmethod
    def content_compile(self, content):
        pass
    def build_compile(self):
        self.header_compile()
        self.content_compile(any)
        self.footer_compile()
#клас та методи для реалізації template method
```

У ньому основні етапи створення сторінки за метод для білду сторінки. У класах CSSRenderer, HTMLParser, ImageLoader та JSInterpreter було імплементовано вище наведений клас та імплементовано метод для компіляції контенту:

```
    def content_compile(self, css_content):
        self.render_css(css_content)
```

```
    def content_compile(self, html_content):
        self.parse(html_content)
```

```
    def content_compile(self, image_url):
        self.load_image(image_url)
```

```
    def content_compile(self, js_code):
        self.execute_js(js_code)
```

У більш детальній реалізації може бути більш детально розписано дії в межах цього методу та імплементовано/задіяно інші методи абстрактного класу.

У класі MainWindow було створено відповідні методи роботи з різним контентом

```
new *
def css_content(self, content):
    return CSSRenderer.CSSRenderer.content_compile(content)
new *
def html_content(self, content):
    return HTMLParser.HTMLParser.content_compile(content)
new *
def image_content(self, content):
    return ImageLoader.ImageLoader.content_compile(content)
new *
def js_content(self, content):
    return JSInterpreter.JSInterpreter.content_compile(content)

#приклад методів для обробки відповідного контенту та його повернення
```

При необхідності роботи з поверненим контентом можуть бути задіяні інші методи абстрактного класу для побудови сторінки, при існуванні на сторінці кількох видів контенту є можливість комбінації методів та їх реалізації для роботи із комбінованим контентом.

Висновок: на цій лабораторній роботі я познайомився з шаблоном template method, засвоїв знання на практиці, продовжив розробку проєкту.