



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

ШАБЛОН «MEDIATOR», «FACADE»,
«BRIDGE», «TEMPLATE METHOD»

Варіант 11

Виконала
студентка групи ІА – 14:
Літвін Юлія

Перевірив:
Мягкий М. Ю.

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів та їх взаємодій для досягнення конкретних функціональних можливостей.
3. Застосування одного з розглянутих шаблонів при реалізації програми.

Варіант:

..11 Web crawler (proxy, chain of responsibility, memento, template method, composite, p2p)

Веб-сканер повинен вміти розпізнавати структуру сторінок сайту, переходити за посиланнями, збирати необхідну інформацію про зазначений термін, видаляти не семантичні одиниці (рекламу, об'єкти javascript і т.д.), зберігати знайдені дані у вигляді структурованого набору html файлів вести статистику відвіданих сайтів і метадані.

Хід роботи

Паттерн Template Method - це шаблон проектування, який визначає скелет алгоритму у батьківському класі, але дозволяє підкласам перевизначити конкретні кроки алгоритму без зміни його структури.

Клас BaseForumScraper:

download_results_file, create_memento, set_memento, _update_parser_stats, get_parser_stats, undo_scrape: Загальні методи, які використовуються всіма підкласами.

Клас DouForumScraper:

Цей клас перевизначає конкретні кроки шаблонного методу:

_handler_scrape_comments_count: Реалізує конкретну логіку отримання кількості коментарів.

_handler_scrape_author_info: Реалізує конкретну логіку отримання інформації про автора.

`_handler_save_to_html`: Реалізує конкретну логіку збереження даних у файл HTML.

`_handler_scrape_page`: Реалізує конкретну логіку скрапінгу сторінки з повідомленнями.

`_handler_scrape_forum`: Реалізує конкретну логіку скрапінгу всього форуму.

Отже, паттерн Template Method забезпечує загальну структуру для процесу веб-скрапінгу, дозволяючи підкласам налаштовувати конкретні кроки.

```
class BaseForumScraper:
    def __init__(self, base_url, headers):
        self.base_url = base_url
        self.headers = headers
        self.all_posts_data = []
        self.stats = ParserStatistics()
        self.caretaker = Caretaker()
        self.current_page = 1

    6 usages (4 dynamic)
    def download_results_file(self, filename, download_path):
        result_file_path = os.path.join('../results/', filename)
        destination_path = os.path.join(download_path, filename)

        try:
            shutil.copy(result_file_path, destination_path)
            return f"File '{filename}' downloaded successfully to '{download_path}'"
        except FileNotFoundError:
            return f"Error: File '{filename}' not found on the server."

    1 usage
    def create_memento(self):
        state = {
            'current_page': self.current_page,
        }
        return Memento(state)

    1 usage
    def set_memento(self, memento):
        self.current_page = memento.state['current_page']

    1 usage
    def _update_parser_stats(self, url):
        self.stats.log_parser_call(url)
```

6 usages (4 dynamic)

```
def get_parser_stats(self):  
    return self.stats.get_parser_stats()
```

4 usages (2 dynamic)

```
def undo_scrape(self):  
    if len(self.caretaker.mementos) > 1:  
        self.caretaker.mementos.pop()  
        previous_memento = self.caretaker.mementos[-1]  
        self.set_memento(previous_memento)
```