



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 №6
ШАБЛони «Abstract Factory»,
«Factory Method», «Memento»,
«Observer», «Decorator»

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 Memento - входить до групи паттернів з управління станом. Його основна ідея полягає в тому, щоб зберігати та відновлювати внутрішній стан об'єкта без розкриття деталей його реалізації.

Клас Memento (Memento):

Представляє стан класу BaseForumScraper, який потрібно зберегти.

Має атрибут state, щоб зберегти фактичний стан.

Клас Caretaker (Caretaker):

Відповідає за відстеження кількох мemento.

Має список з назвою mementos, щоб зберігати екземпляри класу Memento.

Надає методи для додавання нового мemento (add_memento) та отримання мemento за індексом (get_memento).

Клас BaseForumScraper:

Патерн Memento використовується для відстежування та відновлювання стану цього класу.

Має методи для створення мементо (`create_memento`), встановлення його стану з мементо (`set_memento`) та відміни попереднього стану (`undo_scrape`).

Використання:

Перед обходом кожної сторінки створюється нове мементо та додається до утримувача (`self.caretaker.add_memento(self.create_memento())`).

Якщо під час парсингу станеться помилка і сервер перестав відповідати, то під час наступного запуску можливо буде відновити парсинг з тієї сторінки форуму, де код завершив виконання.

Якщо потрібно відмінити операцію, можна викликати метод `undo_scrape` для повернення до попереднього стану.

У підсумку, паттерн Memento класу `BaseForumScraper` зберігати свій стан в різні моменти часу, а клас `Caretaker` відстежує ці стани. Метод `undo_scrape` використовується для повернення стану скрепера до попереднього моменту, що дозволяє використовувати механізм відміни.

```
1 usage
class Memento:
    def __init__(self, state):
        self.state = state
```

```
class Caretaker:
    def __init__(self):
        self.mementos = []
1 usage
    def add_memento(self, memento):
        self.mementos.append(memento)

    def get_memento(self, index):
        return self.mementos[index]
```

```
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']
```

```
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)
```