



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»

Виконала  
студентка групи ІА – 13:  
Майданюк Анастасія

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.

## Хід роботи

Знімок - це патерн проектування, який дозволяє зберігати та відновлювати попередній стан об'єктів, не розголошуючи деталей їхньої внутрішньої реалізації.

Клас **Memento**:

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

Клас `ScraperTemplate` містить два методи у своєму абстрактному визначенні: `create_memento` і `set_memento`. Метод `create_memento` служить для створення нового екземпляра об'єкта `Memento`, який зберігає поточний стан об'єкта. Важливо відзначити, що це глибока копія, щоб уникнути використання посилань. З іншого боку, метод `set_memento` встановлює стан об'єкта на підставі переданого об'єкта `Memento`.

```
1 usage  
class ScraperTemplate(ABC):  
    def create_memento(self):  
        return Memento(copy.deepcopy(self.__dict__))  
  
    def set_memento(self, memento):  
        self.__dict__.update(memento.state)
```

## Клас JobScraper:

Клас JobScraper розширює клас ScraperTemplate і перевизначає методи create\_memento та set\_memento. Ці методи виконують ті самі функції, що і в базовому класі, захоплюють та відновлюють стан об'єкта JobScraper.

```
class JobScraper(ScraperTemplate):
    def __init__(self, base_url, keyword, num_pages=1):
        self.base_url = base_url
        self.keyword = keyword
        self.num_pages = num_pages
        self.proxy_requests = ProxyRequests()
        self.writers = [HTMLWriter]

    1 usage
    def create_memento(self):
        return Memento(copy.deepcopy(self.__dict__))

    1 usage
    def set_memento(self, memento):
        self.__dict__.update(memento.state)
```

У методі scrape\_jobs перед обробкою кожної сторінки створюється новий екземпляр об'єкта Memento, який потім додається до списку mementos. Патерн Memento використовується для зберігання стану об'єкта, надаючи можливість повернутися до його попереднього стану за необхідності. У даному випадку він застосовується для захоплення стану об'єкта JobScraper перед обробкою кожної сторінки і подальшого його відновлення.

Висновок: В ході вивчення патерну проектування "Знімок" (Memento) та його впровадження у класі ScraperTemplate, було розкрито його основні принципи та функціональність. Патерн "Знімок" дозволяє зберігати та відновлювати попередній стан об'єктів, при цьому не розголошуючи деталей їхньої внутрішньої реалізації.