



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 №8

ШАБЛОНИ «COMPOSITE»,  
«FLYWEIGHT», «INTERPRETER»,  
«VISITOR»

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 і 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

Компонувальник — це структурний патерн проектування, що дає змогу згрупувати декілька об'єктів у деревоподібну структуру, а потім працювати з нею так, ніби це одиничний об'єкт.

**ScraperComponent** - це базовий клас зі спільними методами, а

**CompositeScraper** - конкретна реалізація.

**ScraperComponent** (абстрактний базовий клас):

Визначає спільний інтерфейс для всіх класів(download\_results\_file, get\_parser\_stats, scrape\_forum, undo\_scrape).

**CompositeScraper:**

Він містить список об'єктів ScraperComponent і реалізує спільні методи, делегуючи виклики своїм підкласам.

Патерн Composite використовується в класі CompositeScraper, що дозволяє обробляти один скрепер або колекцію скрейперів у єдиний спосіб. Це може бути корисно, коли потрібно виконати операції над декількома скрейперами так, ніби вони були єдиним об'єктом.

```
class ScraperComponent(ABC):
    4 usages (4 dynamic)
    @abstractmethod
    def download_results_file(self, filename, download_path):
        pass

    4 usages (4 dynamic)
    @abstractmethod
    def get_parser_stats(self):
        pass

    3 usages (3 dynamic)
    @abstractmethod
    def scrape_forum(self, output_filename, tag, start_date=None, end_date=None):
        pass

    2 usages (2 dynamic)
    @abstractmethod
    def undo_scrape(self):
        pass
```

```
1 usage
class CompositeScraper(ScraperComponent):
    def __init__(self):
        self.scrapers = []

    1 usage
    def add_scraper(self, scraper):
        self.scrapers.append(scraper)

    4 usages (4 dynamic)
    def download_results_file(self, filename, download_path):
        for scraper in self.scrapers:
            scraper.download_results_file(filename, download_path)

    4 usages (4 dynamic)
    def get_parser_stats(self):
        for scraper in self.scrapers:
            scraper.get_parser_stats()

    3 usages (3 dynamic)
    def scrape_forum(self, output_filename, tag, start_date=None, end_date=None):
        for scraper in self.scrapers:
            scraper.main_handler('main', output_filename, tag, start_date=start_date, end_date=end_date)

    2 usages (2 dynamic)
    def undo_scrape(self):
        for scraper in self.scrapers:
            scraper.undo_scrape()
```