



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»

Виконала
студентка групи ІА – 13:
Майданюк Анастасія

Перевірив:
Мягкий М.Ю.

Київ 2023

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

Патерн "Компонувальник" є структурним патерном дизайну, який дозволяє організувати багато об'єктів у деревовидну структуру, де вони можуть бути оброблені як єдиний елемент.

У контексті паттерну "Composit", він використовується для створення єдиного об'єкту CompositeWriter, який об'єднує різні типи "письменників".

CompositeWriter: Цей клас функціонує як контейнер, що включає в себе інші об'єкти типу "письменник", наприклад HTMLWriter. Він забезпечує функціонал для додавання нових "письменників", а також методи для виконання загальних операцій, таких як запис заголовків або підписів, розповсюджуючи ці дії на всі включені об'єкти.

HTMLWriter: Цей клас є прикладом конкретного "листа" у структурі композиту, який виконує певні функції пов'язані з форматуванням HTML.

JobScraper: Цей клас використовує композит для злиття різних "письменників" разом і використовує їх для збору та виведення даних. В

процесі методу `scrape_jobs` клас переглядає різні сторінки, збирає інформацію про вакансії і використовує `composite_writer` для запису цих даних у вихідний документ.

```
1 class CompositeWriter:
2     def __init__(self, writers=None):
3         self.writers = writers or []
4
5     def add_writer(self, writer):
6         self.writers.append(writer)
7
8     def write_header(self):
9         for writer in self.writers:
10            writer.write_header()
11
12    def write_footer(self):
13        for writer in self.writers:
14            writer.write_footer()
15
16    def write_page_header(self, page):
17        for writer in self.writers:
18            writer.write_page_header(page)
19
20    def write_job_container(self, title, url, details_callback):
21        for writer in self.writers:
22            writer.write_job_container(title, url, details_callback)
23
24    def write(self, content):
25        for writer in self.writers:
26            writer.file.write(content)
```

Висновок:

У ході виконання лабораторної роботи особливу увагу я зосередила на паттерні "Composite", що дозволило ефективно організувати об'єкти у деревовидну структуру та виконувати операції над ними як над єдиним цілим. Використання класу `CompositeWriter`, як основного елемента цієї структури, дало можливість інтегрувати різні типи "письменників", таких як `HTMLWriter`, що значно полегшило процес форматування та запису даних.