

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**  
із дисципліни «*Технології розробки програмного забезпечення*»  
Тема: «*Патерни проектування*»

**Виконав:**

Студент групи IA-34  
Ястремський Богдан

**Перевірив:**

асистент кафедри ІСТ  
Мягкий Михайло Юрійович

**Тема:** Патерни проектування.

**Мета:** Вивчити структуру шаблонів «Mediator», «Facade», «Bridge», «Template method» та навчитися застосовувати їх в реалізації програмної системи..

**Застосунок (№6):** Web-browser (proxy, chain of responsibility, factory method, template method, visitor, p2p).

Веб-браузер повинен мати можливість зробити наступне: мати адресний рядок для введення адреси сайту, переміщатися і відображати структур html документа, переглядати підключений javascript та css файли, перегляд всіх підключених ресурсів (зображень), коректна обробка відповідей з сервера (коди відповідей HTTP) – переходи при перенаправленнях, відображення сторінок 404 і 502/503.

### **Короткі теоретичні відомості:**

Шаблон «Mediator» (посередник) використовується для визначення взаємодії об'єктів за допомогою іншого об'єкта (замість зберігання посилань один на одного). Даний шаблон схожий на шаблон «команда», проте в даному випадку замість зберігання даних про конкретну дію, зберігаються дані про взаємодії між компонентами.

Шаблон «Facade» (фасад) передбачає створення єдиного уніфікованого способу доступу до підсистеми без розкриття внутрішніх деталей підсистеми. Оскільки підсистема може складатися з безлічі класів, а кількість її функцій – не більше десяти, то щоб уникнути створення «спагетікоду» (коли все тісно пов'язано між собою) виділяють один загальний інтерфейс доступу, здатний правильним чином звертатися до внутрішніх деталей.

Шаблон «Bridge» (міст) використовується для поділу інтерфейсу і його реалізації. Це необхідно у випадках, коли може існувати кілька різних абстракцій, над якими можна проводити дії різними способами.

Шаблон «Template Method» (шаблонний метод) дозволяє реалізувати покроково алгоритм в абстрактному класі, але залишити специфіку реалізації підкласам. Даний шаблон дещо нагадує шаблон «Фабричний метод», однак

область його використання абсолютно інша – для покрокового визначення конкретного алгоритму; більш того, даний шаблон не обов'язково створює нові об'єкти – лише визначає послідовність дій.

### Хід роботи:

- Повна діаграма класів (без шаблону).

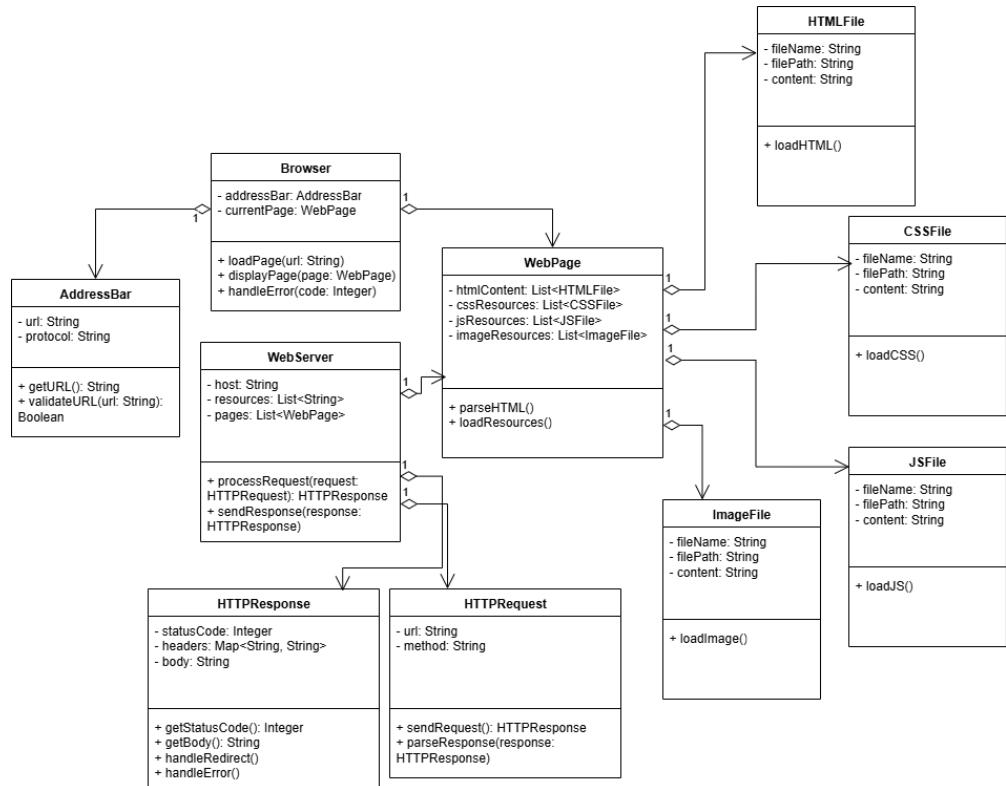


Рис. 7.1 – Діаграма класів системи

- Діаграма класів з використанням шаблону «Template method»:

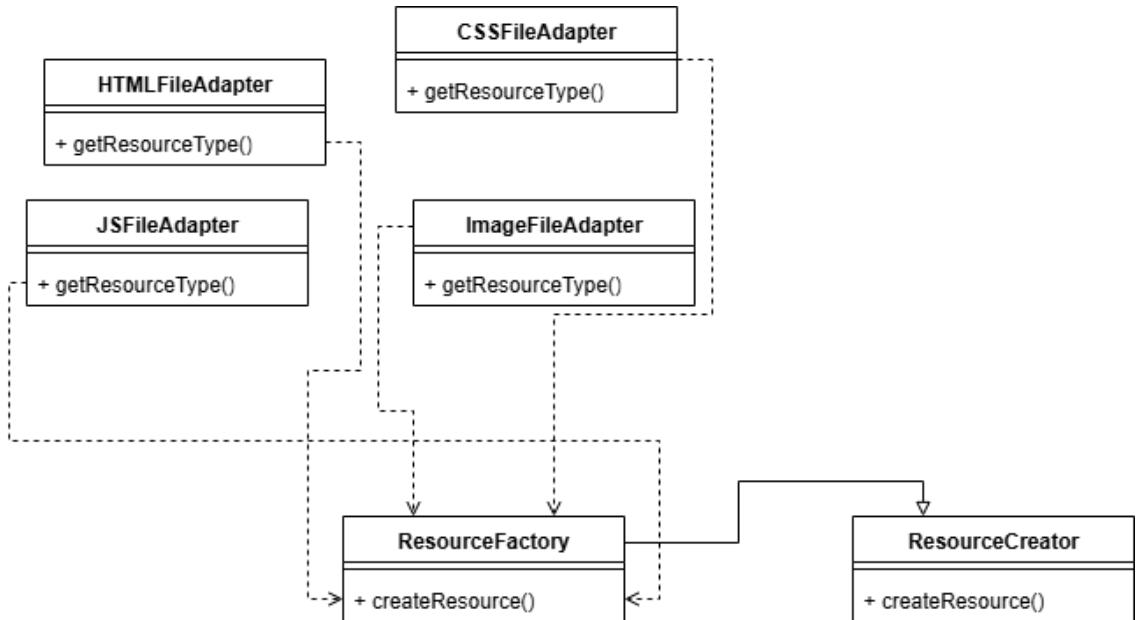


Рис. 7.2 – Діаграма класів системи з використанням шаблону «Template Method»

По суті, це напів-шаблон, оскільки функція абстрактних класів, як раз у перевизначенні методів класів-наслідників. Саме тому цей шаблон є частиною Фабричного методу.

Я визначаю абстрактний клас `ResourceCreator`, який містить в собі метод для створення ресурсів. Далі я визначаю творця-фабрику `ResourceFactory`, який вже буде створювати конкретні продукти-ресурси (які я назвав адаптерами), базуючись на розширеннях.

Далі я для кожного типу файлу створив власний продукт-ресурс, який містить в собі метод для повернення типу ресурсу для подальшої роботи з ним.

На вершині в нас інтерфейс ресурс, який визначає основні методи для роботи з файлами, які потім знадобляться для завантажування ресурсів.

У якості використання патерну, я оновив метод `WebPage.parseHTML()`, який тепер базується тільки на регулярних виразах по перевірці тегів, які віднаходять ті, що треба для обробки. Таким чином, я додаю один із цих ресурсів до загальних ресурсів, не

перевіряючи їх нічим, окрім регулярних виразів, сприошуєчи власне метод, який містив купу умов, а зараз потребує в рази менше коду.

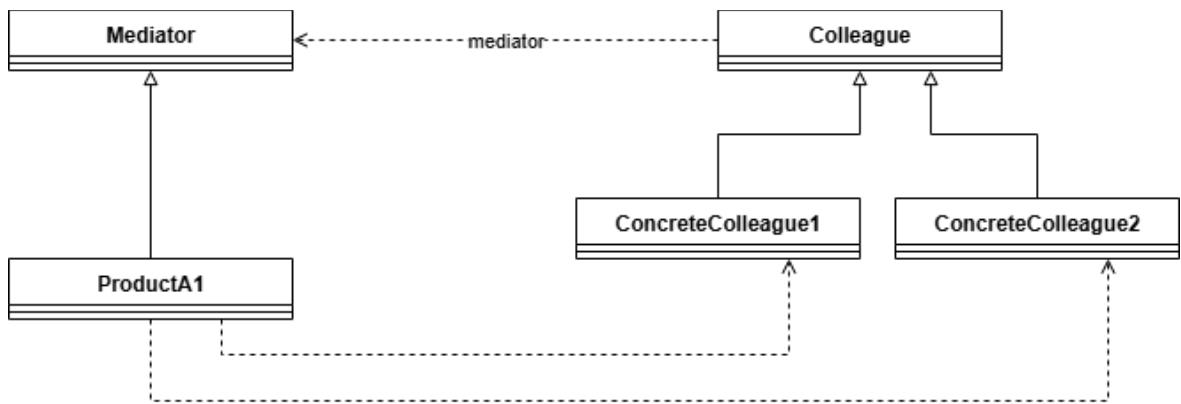
**Висновок:** на даному лабораторному занятті я продовжив знайомством з поняттям патернів проектування, які є їх види, та їхні плюси і мінуси. В якості використованого патерну я обрав «Template Method», та імплементував його як створювач ресурсів для завантаження на сторінку. Я позбавив клас від прив'язки до конкретних класів, використовуючи регулярні вирази як парсер для кожного необхідного мені типу ресурсу. Паралельно з цим, я спростив підтримку коду, виділивши всі виробництва ресурсів в одне місце. Не дивлячись на те, що додати новий тип даних, типу відео, аудіо і т.д. дуже легко в цьому патерні – створивши відповідний ConcreteProduct – ієархія отримується доволі складна, а також цей патерн сильно обмежує розробника скелетом абстрактного класу – саме тому слід гарно продумати, що цей клас буде містити.

## **Відповіді на контрольні питання:**

1. Яке призначення шаблону «Посередник»?

Шаблон «Mediator» (посередник) використовується для визначення взаємодії об'єктів за допомогою іншого об'єкта (замість зберігання посилань один на одного). Даний шаблон схожий на шаблон «команда», проте в даному випадку замість зберігання даних про конкретну дію, зберігаються дані про взаємодії між компонентами.

2. Нарисуйте структуру шаблону «Посередник».



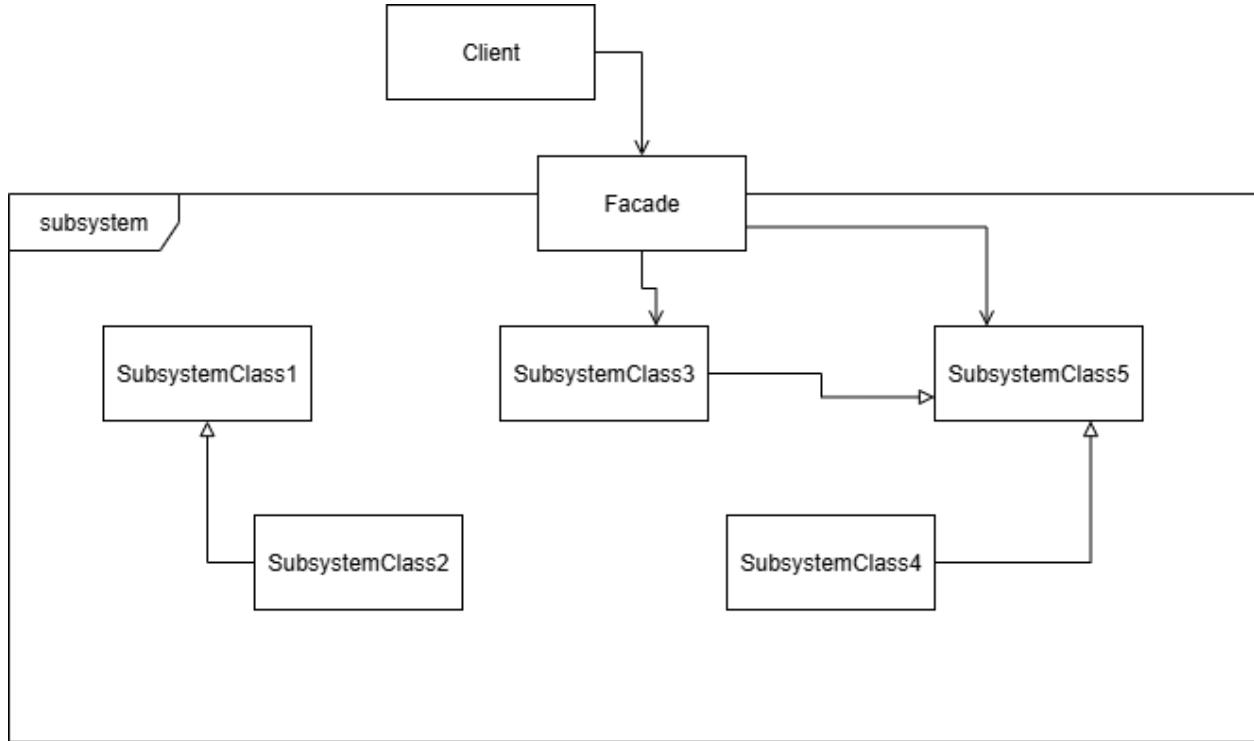
3. Які класи входять в шаблон «Посередник», та яка між ними взаємодія?

- Посередник (Mediator): Визначає інтерфейс для взаємодії між об'єктами Колег. Він оголошує методи для обміну повідомленнями.
- Конкретний посередник (Concrete Mediator, ProductA1): Реалізує інтерфейс Посередника та координує взаємодію між конкретними об'єктами Колег. Він містить логіку, яка визначає, як різні колеги реагують на події один одного.
- Колега (Colleague): Базовий клас або інтерфейс для всіх об'єктів, які беруть участь у взаємодії. Кожен колега знає лише про існування свого Посередника, але нічого не знає про інших колег.
- Конкретний колега (Concrete Colleague 1, Concrete Colleague 2): Конкретна реалізація Колеги.

4. Яке призначення шаблону «Фасад»?

Шаблон «Facade» (фасад) передбачає створення єдиного уніфікованого способу доступу до підсистеми без розкриття внутрішніх деталей підсистеми.

5. Нарисуйте структуру шаблону «Фасад».



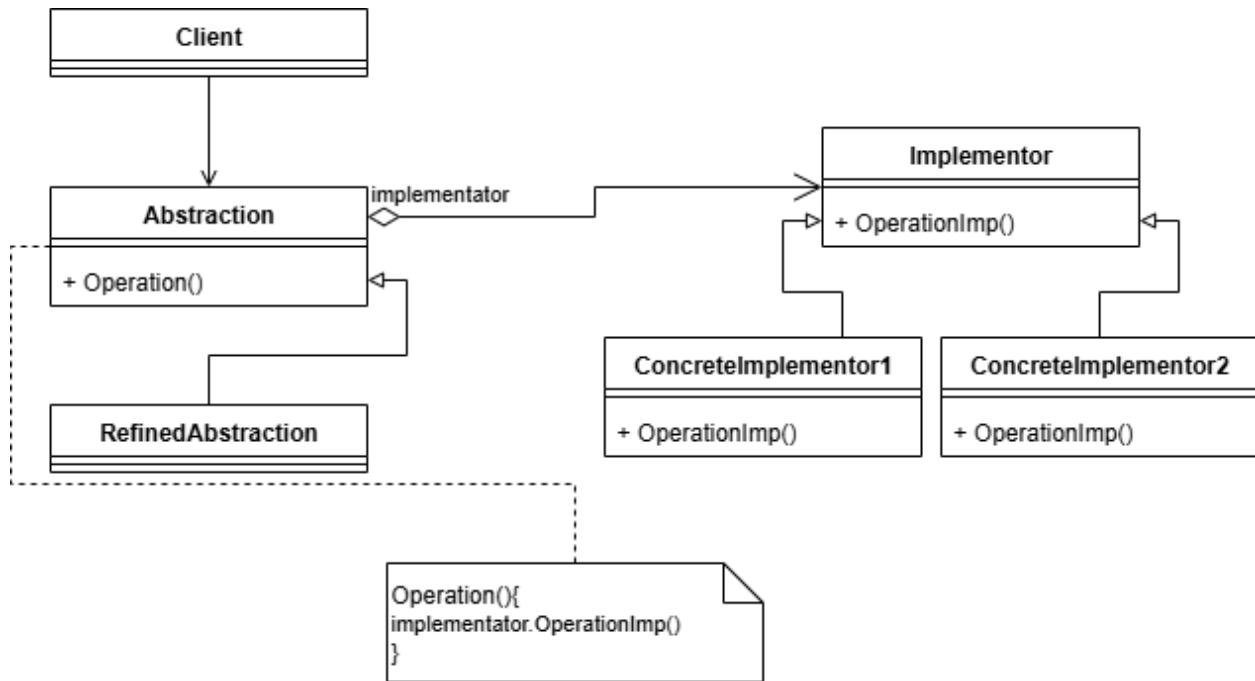
6. Які класи входять в шаблон «Фасад», та яка між ними взаємодія?

- Фасад (Facade): Єдиний клас, який надає спрощений інтерфейс Клієнту. Він приховує складність підсистеми та делегує запити клієнта відповідним об'єктам усередині підсистеми.
- Підсистема (Subsystem Classes): Набір складних, взаємозалежних класів, які виконують реальну роботу.
- Клієнт (Client): Код, який використовує функціональність підсистеми.

7. Яке призначення шаблону «Міст»?

Шаблон «Bridge» (міст) використовується для поділу інтерфейсу і його реалізації. Це необхідно у випадках, коли може існувати кілька різних абстракцій, над якими можна проводити дії різними способами.

8. Нарисуйте структуру шаблону «Міст».



9. Які класи входять в шаблон «Міст», та яка між ними взаємодія?

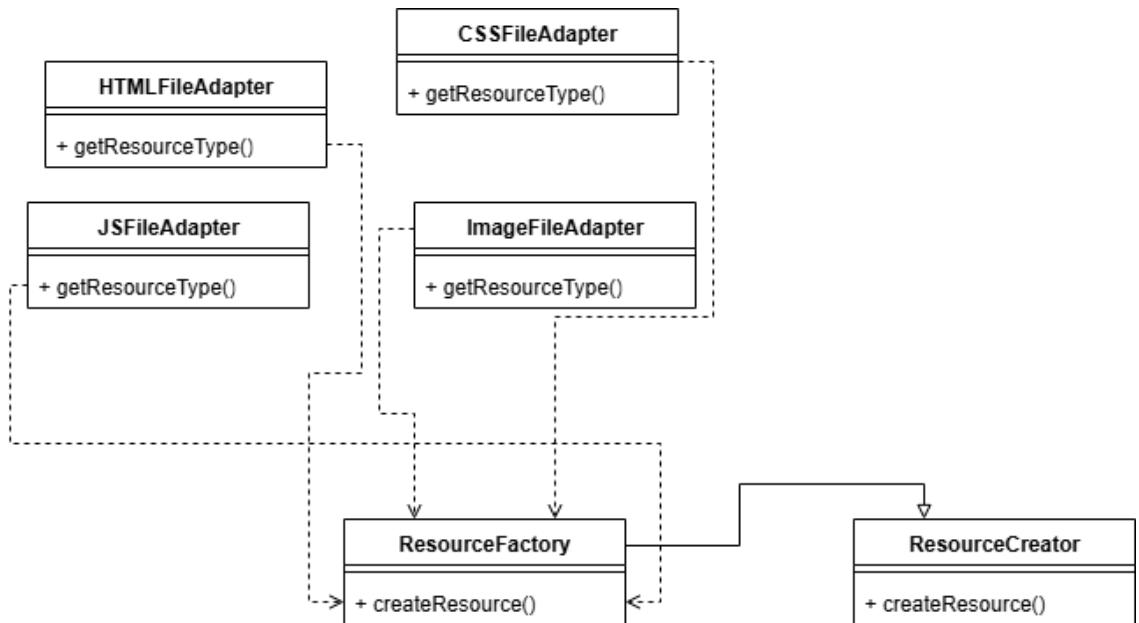
- Абстракція (Abstraction): Визначає інтерфейс високого рівня для клієнтів. Вона містить бізнес-логіку і підтримує посилання (міст) на об'єкт Реалізації.
- Уточнена абстракція (Refined Abstraction): Розширює інтерфейс, визначений Абстракцією, додаючи специфічну логіку або додаткові операції.
- Реалізація (Implementor): Визначає інтерфейс для класів конкретних реалізацій. Він має бути якомога простішим і містити лише примітивні операції, необхідні Абстракції.
- Конкретна реалізація (Concrete Implementor): Реалізує інтерфейс Реалізації.

10. Яке призначення шаблону «Шаблонний метод»?

Шаблон «Template Method» (шаблонний метод) дозволяє реалізувати покроково алгоритм в абстрактному класі, але залишити специфіку реалізації підкласам. Даний шаблон дещо нагадує шаблон «Фабричний метод», однак область його використання абсолютно інша – для покрокового визначення конкретного алгоритму; більш того, даний

шаблон не обов'язково створює нові об'єкти – лише визначає послідовність дій.

11. Нарисуйте структуру шаблону «Шаблонний метод».



12. Які класи входять в шаблон «Шаблонний метод», та яка між ними взаємодія?

- Абстрактний клас / Творець (Abstract Class / Creator): Визначає основний алгоритм у вигляді шаблонного методу. Цей метод містить послідовність кроків.
- Конкретний клас (Concrete Class): Підклас, який реалізує або перевизначає абстрактні методи, визначені в Абстрактному класі.

13. Чим відрізняється шаблон «Шаблонний метод» від «Фабричного методу»?

Шаблонний метод, який по суті не є повноцінним шаблоном, а лише послідовність дій, є частиною фабричного методу.

14. Яку функціональність додає шаблон «Міст»?

- Дозволяє змінювати ієрархії абстракції та реалізації незалежно одна від одної.
- Розділивши абстракцію від реалізації отримуємо більшу гнучкість та простіший супровід такого коду.