



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
з дисципліни **«Технології розроблення програмного
забезпечення»**

на тему «шаблони проектування «mediator», «facade»,
«bridge», «template method» »

Тема для лабораторного циклу «HTTP-сервер»

Виконала

студентка групи ІА-04

Глушко

Юлія Петрівна

Перевірів

Колеснік Валерій

Миколайович

Защищено з балом _____

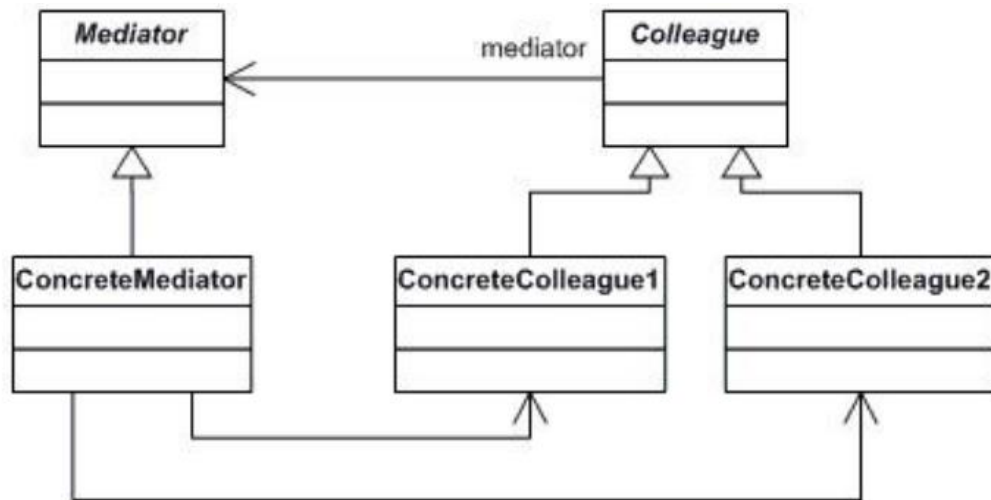
Київ 2022

Хід роботи :

Шаблон «Mediator»

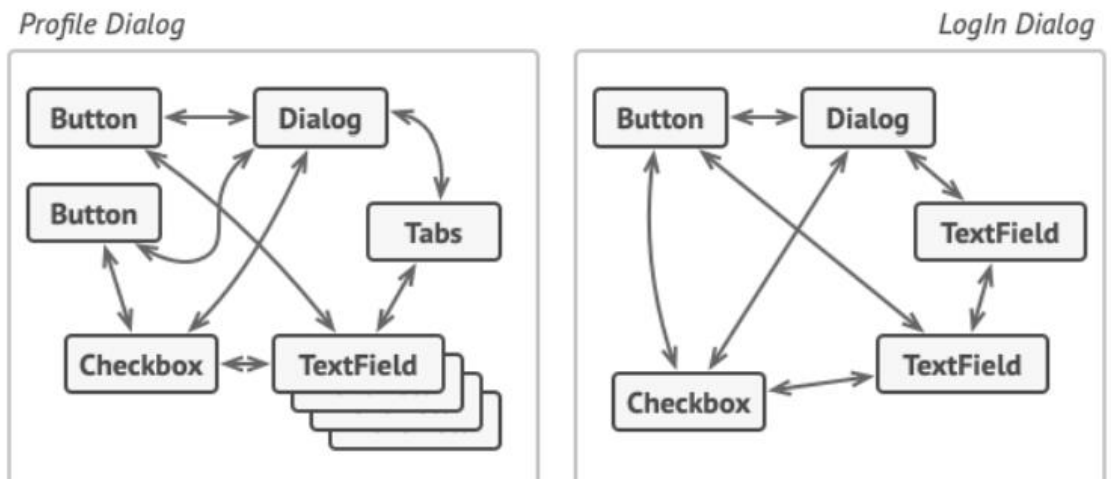
Теоретичні відомості

- Структура :



- Призначення патерну:

Шаблон «Mediator» (посередник) використовується для визначення взаємодії об'єктів за допомогою іншого об'єкта (замість зберігання посилань один на одного).



Даний шаблон зручно застосовувати у випадках, коли безліч об'єктів

взаємодіє між собою деяким структурованим чином, однак складним для розуміння. У такому випадку вся логіка взаємодії виноситься в окремий об'єкт. Кожен із взаємодіючих об'єктів зберігає посилання на об'єкт «медіатор».

Патерн «Mediator» змушує об'єкти спілкуватися не безпосередньо один з одним, а через окремий об'єкт-посередник, який знає, кому потрібно перенаправити той чи інший запит. Завдяки цьому, компоненти системи будуть залежати тільки від посередника, а не від десятків інших компонентів.

Щоб зробити код ще більш гнучким, можна виділити загальний інтерфейс для всіх посередників.

Таким чином, посередник приховує в собі всі складні зв'язки і залежності між класами окремих компонентів програми. А чим менше зв'язків мають класи, тим простіше їх змінювати, розширювати і повторно використовувати.

- Переваги та недоліки:

- + Усуває залежності між компонентами, дозволяючи повторно їх використовувати.

- + Спрощує взаємодію між компонентами.

- + Централізує управління в одному місці.

- Посередник може сильно роздуватися.

Застосування при реалізації лабораторного циклу

Оскільки сервер повинен бути розроблений за Р2Р архітектурою, патерн «Mediator» є дуже доречним. Його було використано при реалізації взаємодії користувачів. Тобто коли якийсь певний користувач відправляє дані на сервер, вони автоматично відправляються іншим підключеним користувачам.

До серверу можуть бути підключеними багато користувачів. На кожне підключення створюється екземпляр класу `HTTPConnection`, що відповідає за це підключення. Кожен підключений користувач може відправляти дані іншим підключеним користувачам та отримувати дані від них.

Для обміну даними був створений клас `HTTPConnectionListener` інтерфейсу `ConnectionListener`. В цьому класі існує список всіх активних підключень та методи для додавання / видалення підключення і розсилки повідомлень на активні підключення. Якщо з'являється нове підключення, воно звертається до цього класу, щоб стати доданим до списку підключень, а після втрати з'єднання, щоб бути видаленим.

Якщо користувач бажає розіслати повідомлення, він звертається до `HTTPConnectionListener`, викликавши його метод `sendMessage`

```
@Override
public synchronized void sendMessage(HTTPConnection
httpConnection, Map<String, String> data) {
    sendAllConnections(httpConnection, data);
}
```

```
private void sendAllConnections( HTTPConnection
httpConnection, Map<String, String> data) {
    LOGGER.info(connections.size() + " connections size ");
    for (HTTPConnection connection : connections)
        if (! (connection==httpConnection))
            connection.getData(data);
}
```

Яким чином користувач буде отримувати повідомлення, описано в методі `getData` класу `HTTPConnection`. У нашому випадку будується відповідь від серверу, відповідно до ситуації та даних.

Діаграму класів, що реалізують даний патерн, можна побачити на рисунку 1.

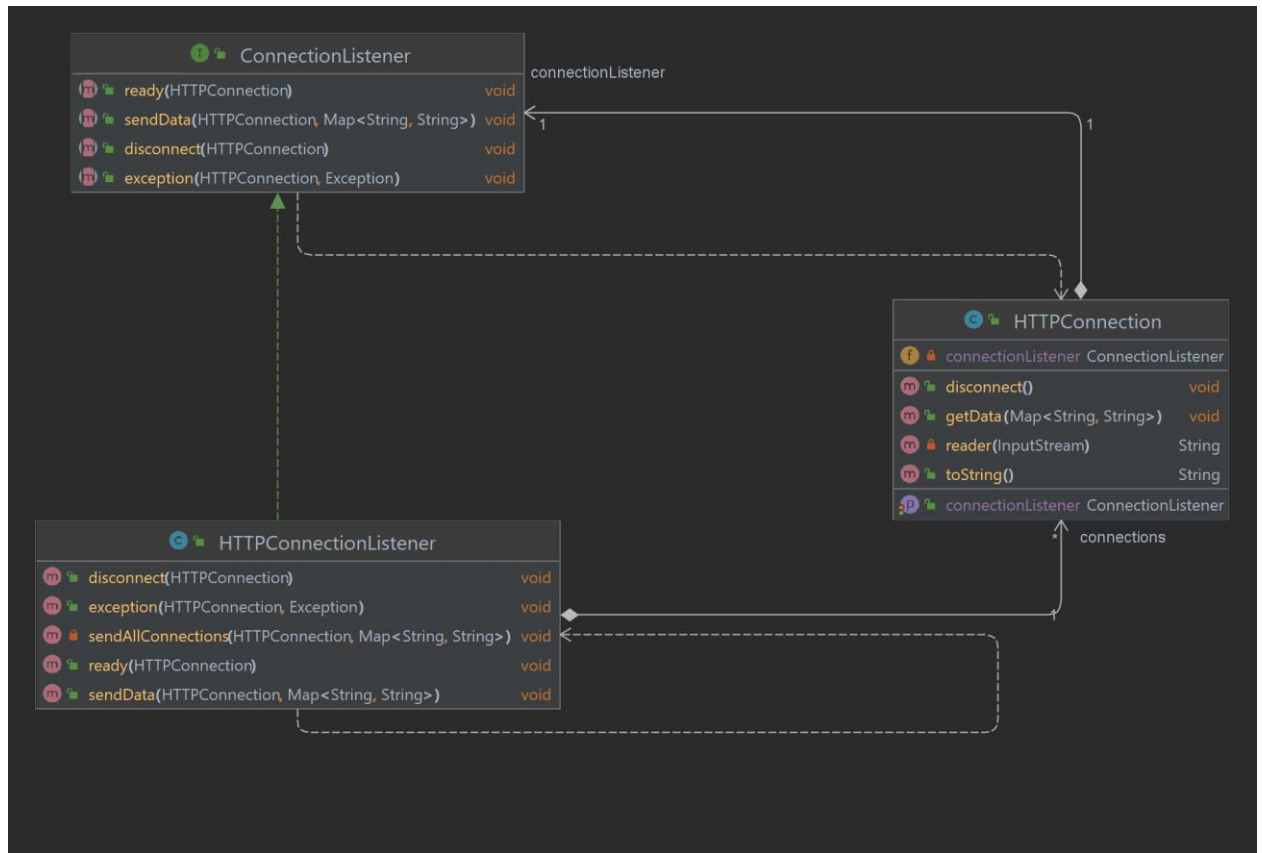


Рис.1 – діаграма класів, що реалізують шаблон «Mediator»

Висновок: під час виконання лабораторної роботи було розглянуто патерни проектування «mediator», «facade», «bridge», «template method». Патерн «mediator» був застосований як частина структури лабораторного практикуму