



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

Технології розроблення програмного забезпечення

«Патерни проектування.»

Тема: IRC-Клієнт

Виконала:

студентка групи IA-32

Красоха В.О.

Перевірив:

Мягкий М.Ю.

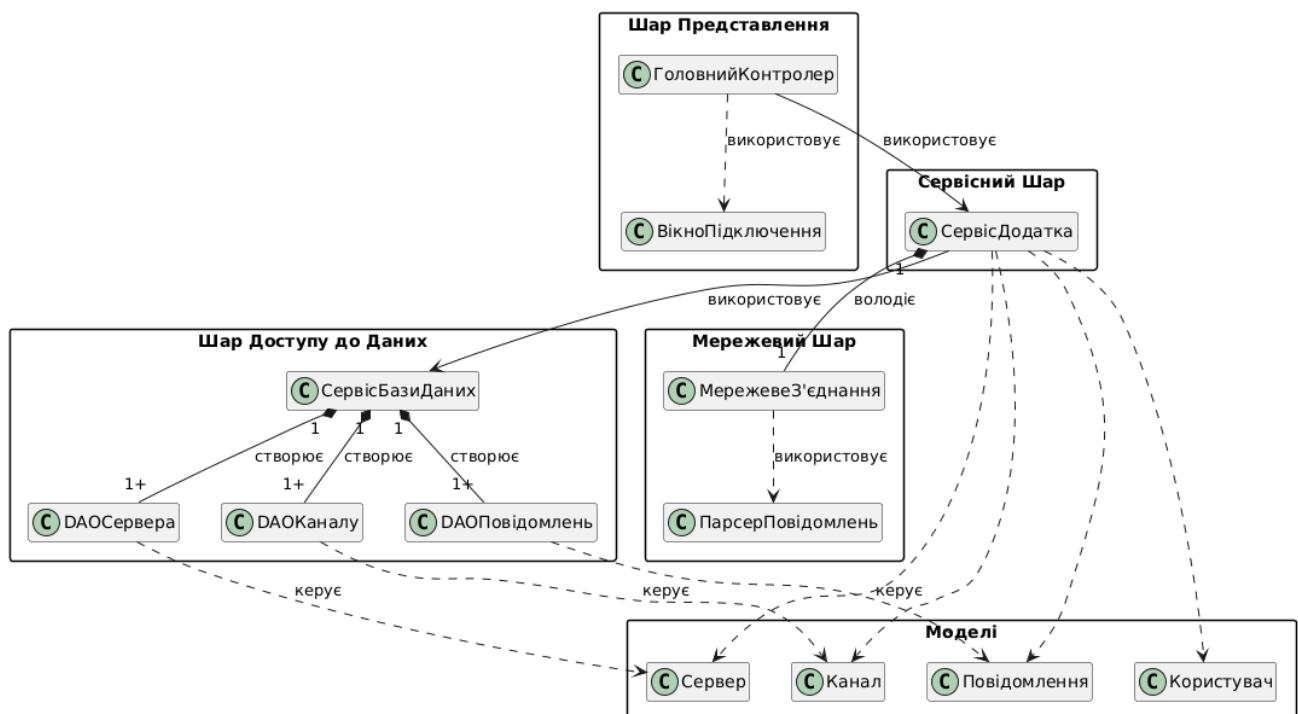
Київ 2025

Тема: Патерни проектування.

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

Хід роботи

Діаграма класу:



Шаблон Facade (Фасад)

Це структурний патерн, який надає простий інтерфейс до складної системи класів, бібліотек або фреймворків. У проєкті знаходиться в класі IRCService (або СервісДодатка). Програма має складні підсистеми: роботу з сокетами (IRCConnection), парсинг тексту (MessageParser) та роботу з базою даних (MessageDao). Замість того, щоб MainController керував сокетами та SQL-запитами напряму, він просто викликає один метод сервісу, наприклад connect() або sendMessage(). IRCService ховає («за фасадом») всю складність і взаємодіє з іншими об'єктами за контролера.

Шаблон Mediator (Посередник)

Це поведінковий патерн, який забезпечує взаємодію багатьох об'єктів без необхідності посилатися один на одного напряму. Він сприяє послабленню

зв'язності. Знову ж таки, роль посередника виконує MainController у поєднанні з IRCSERVICE. Об'єкти графічного інтерфейсу (кнопки, текстові поля) не знають про існування бази даних або мережевого сокета. Коли натискається кнопка, вона повідомляє Контролер, а той виступає «диригентом». Він каже мережевому модулю: «відправ», базі даних: «збережи», а вікну чату: «відобрази». На діаграмі послідовності чітко видно цей централізований вузол керування.

Доданий код:

Main.java:

```
package org ircclient;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

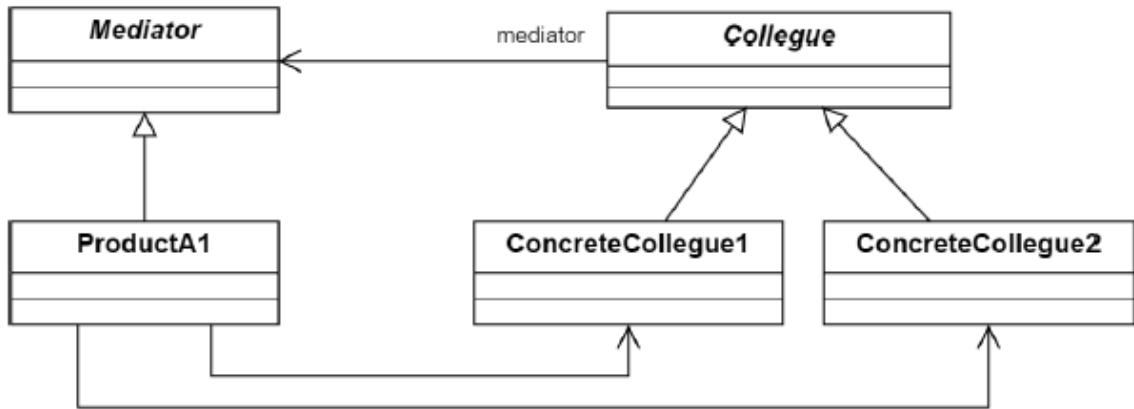
import java.io.IOException;

/**
 * Головний клас додатку IRC клієнта
 */
public class Main extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(Main.class.getResource("/fxml/main.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), 1000, 700);
        scene.getStylesheets().add(getClass().getResource("/css/styles.css").toExternalForm());
        stage.setTitle("IRC Клієнт");
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```

Контрольні питання:

1. Яке призначення шаблону «Посередник»?
Шаблон Mediator (Посередник) призначений для організації взаємодії між об'єктами через централізований об'єкт-посередник, щоб зменшити залежності між компонентами.
2. Нарисуйте структуру шаблону «Посередник».



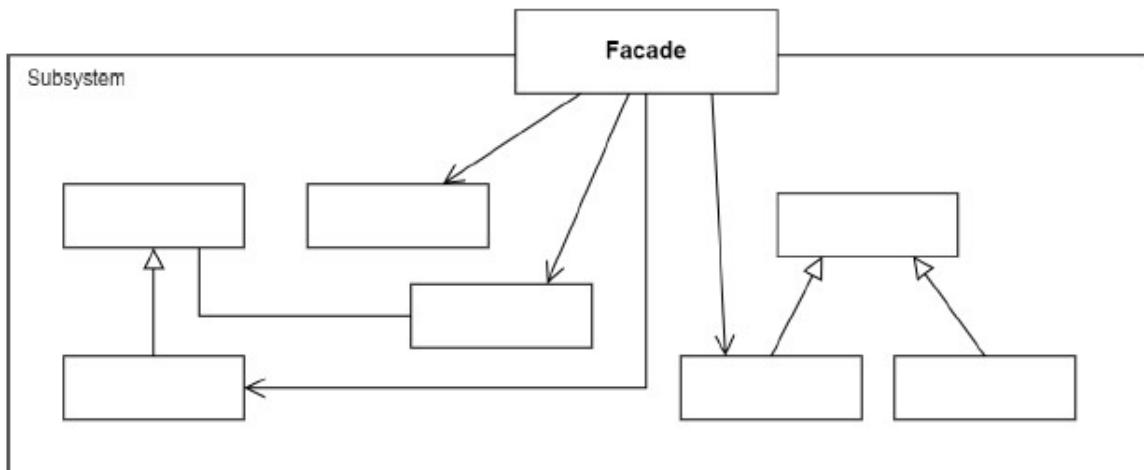
3. Які класи входять в шаблон «Посередник», та яка між ними взаємодія?

- Mediator — інтерфейс посередника
- ConcreteMediator — реалізація, координує взаємодію колег
- Colleague — об'єкти, які взаємодіють через посередника
Colleague звертається до ConcreteMediator, а він передає повідомлення іншим Colleague.

4. Яке призначення шаблону «Фасад»?

Шаблон Facade (Фасад) надає спрощений інтерфейс до складної підсистеми, приховуючи її внутрішню складність від клієнта.

5. Нарисуйте структуру шаблону «Фасад».



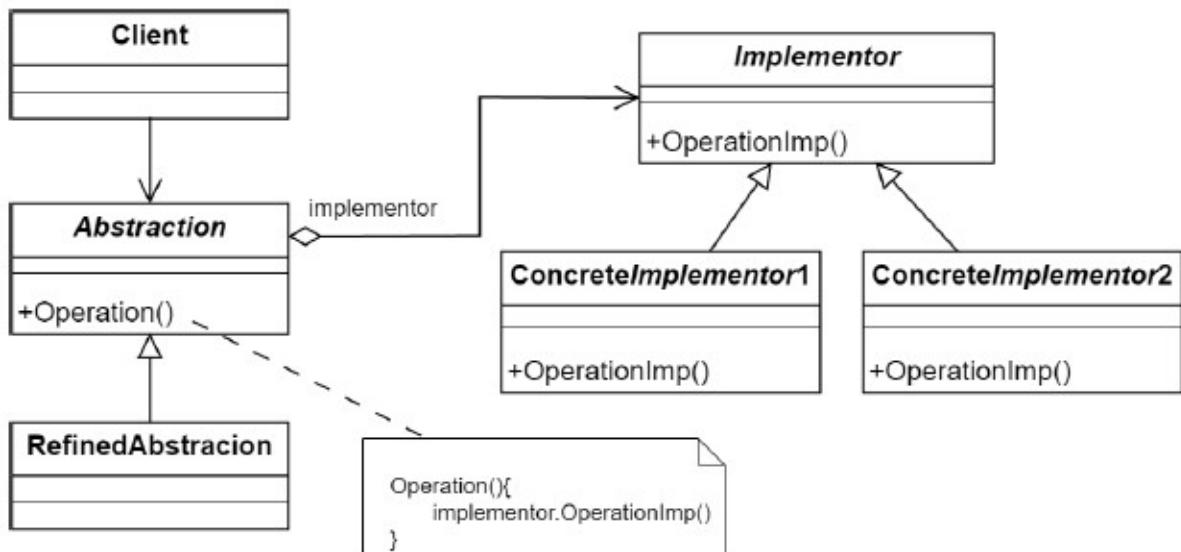
6. Які класи входять в шаблон «Фасад», та яка між ними взаємодія?

- Facade — спрощений інтерфейс до підсистеми
 - Subsystem — класи складної підсистеми
- Facade делегує виклики клієнта відповідним об'єктам Subsystem.

7. Яке призначення шаблону «Міст»?

Шаблон Bridge (Міст) дозволяє розділити абстракцію та реалізацію, щоб вони могли змінюватися незалежно одна від одної.

8. Нарисуйте структуру шаблону «Міст».



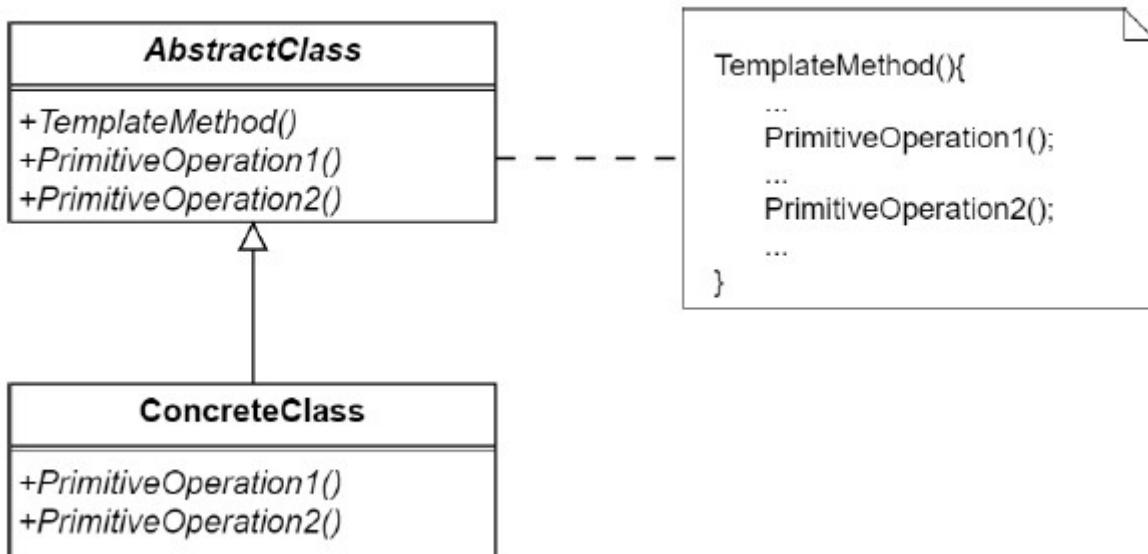
9. Які класи входять в шаблон «Міст», та яка між ними взаємодія?

- Abstraction — абстракція, містить посилання на Implementor
 - RefinedAbstraction — розширена абстракція
 - Implementor — інтерфейс реалізації
 - ConcreteImplementor — конкретна реалізація
- Abstraction делегує виконання методів ConcreteImplementor через Implementor.

10. Яке призначення шаблону «Шаблонний метод»?

Шаблон Template Method (Шаблонний метод) визначає скелет алгоритму у базовому класі, залишаючи реалізацію деяких кроків підкласам.

11. Нарисуйте структуру шаблону «Шаблонний метод».



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

- **AbstractClass** — містить шаблонний метод та абстрактні кроки алгоритму
- **ConcreteClass** — реалізує абстрактні кроки алгоритму
Клієнт викликає шаблонний метод **AbstractClass**, який делегує виконання конкретних кроків **ConcreteClass**.

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

- Шаблонний метод визначає скелет алгоритму і делегує реалізацію кроків підкласам
- Фабричний метод зосереджений на створенні об'єкта, делегуючи підкласам вибір конкретного класу

Тобто Template Method контролює алгоритм, а Factory Method — конкретний об'єкт.

14. Яку функціональність додає шаблон «Міст»?

Bridge дозволяє незалежно змінювати абстракцію та реалізацію, робить код гнучким і спрощує розширення системи без модифікації існуючих класів.

Висновок: Під час виконання лабораторної роботи я вивчила структуру шаблонів «Mediator», «Facade», «Bridge», «Template method» та навчилася застосовувати їх в реалізації програмної системи.