

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»
Тема: «Патерни проектування»

Виконав:
студент групи ІА-32
Задорожний Костянтин
Леонідович

Перевірив:
Мягкий Михайло
Юрійович

Київ - 2025

Тема: Патерни проектування.

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

Завдання

- Ознайомитись з короткими теоретичними відомостями.
- Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.
- Реалізувати один з розглянутих шаблонів за обраною темою.
- Реалізувати не менше 3-х класів відповідно до обраної теми.
- Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен містити: діаграму класів, яка представляє використання шаблону в реалізації системи, навести фрагменти коду по реалізації цього шаблону.

Предметна область:

22. FTP-server (state, builder, memento, template method, visitor, client-server)
FTP-сервер повинен вміти коректно обробляти і відправляти відповіді п протоколу FTP, з можливістю створення користувачів (з пароллями) і доступних їм папок, розподілу прав за стандартною схемою (rwe), ведення статистики з'єднань, обмеження максимальної кількості підключень і максимальної швидкості поширення глобально і окремо для кожного облікового запису.

Хід роботи

Для виконання цієї лабораторної роботи я використав патерн: «Шаблонний метод» (Template Method).

Цей патерн підходить для реалізації алгоритмів, які мають спільну структуру, але відрізняються в деталях реалізації окремих кроків.

У контексті FTP-сервера можна використати цей патерн для пост-обробки завантажених файлів.

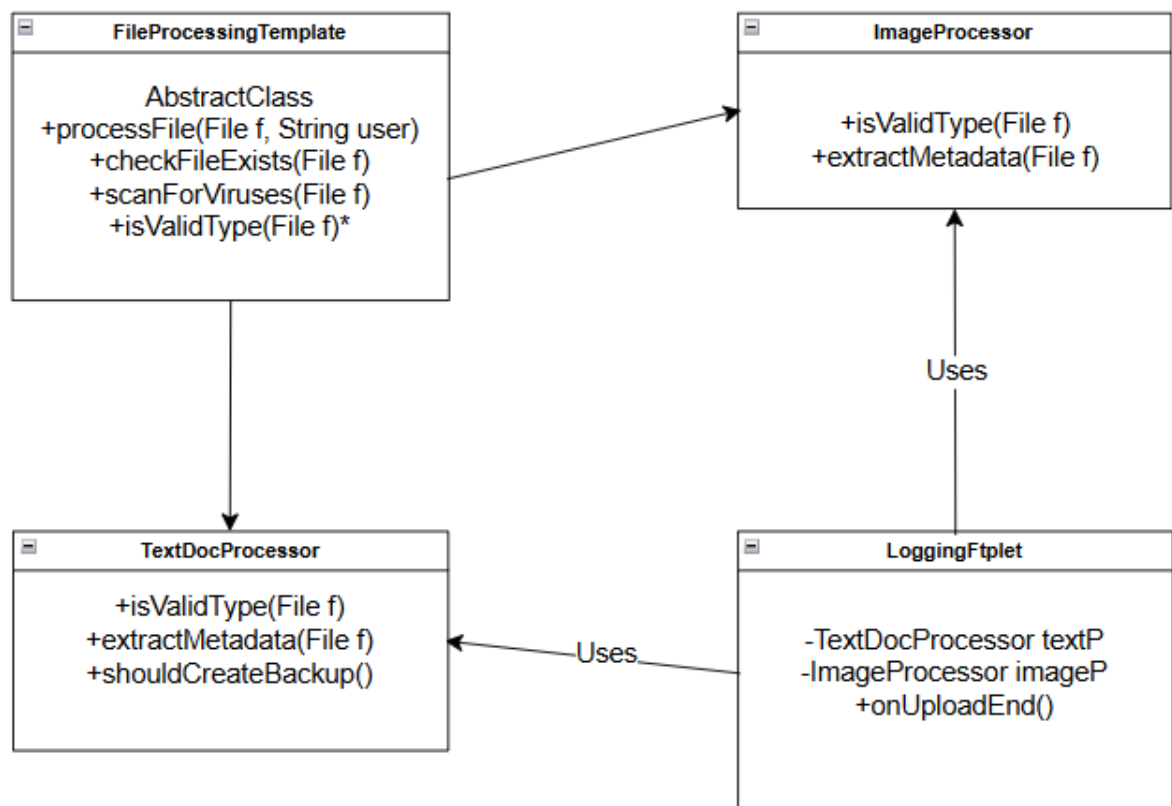
Спільний алгоритм (скелет): 1. Перевірка імені файлу -> 2. Перевірка на віруси (імітація) -> 3. Специфічна обробка (залежить від типу) -> 4. Створення резервної копії (хук) -> 5. Логування.

Структура:

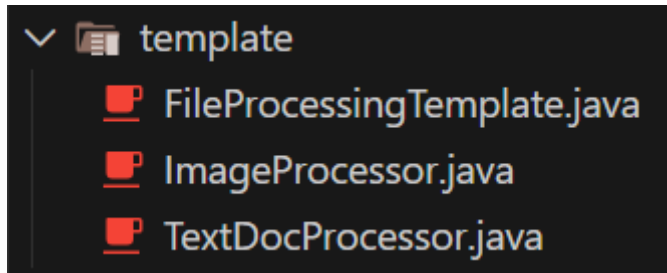
AbstractClass: Оголошує методи, які виступають кроками алгоритму, а також сам шаблонний метод, який викликає ці кроки у певному порядку.

ConcreteClass: Реалізує абстрактні кроки алгоритму, специфічні для конкретного підкласу.

Діаграма структури патерну Template Method для FTP server:



Реалізація в коді:



`FileProcessingTemplate` – абстрактний клас, що визначає структуру алгоритму обробки файлу. Він гарантує, що кроки виконуватимуться у суворо визначеному порядку, який не можна порушити в підкласах.

```
public abstract class FileProcessingTemplate {

    public final void processFile(File file, String uploaderName) {
        System.out.println("--- [TEMPLATE START] Processing " + file.getName() + " ---");

        if (!checkFileExists(file)) {
            System.out.println(x: "File not found!");
            return;
        }

        if (isValidType(file)) {
            scanForViruses(file);

            extractMetadata(file);

            if (shouldCreateBackup()) {
                createBackup(file);
            }

            sendNotification(uploaderName, file.getName());
        } else {
            System.out.println(x: "Skipping processing: Invalid file type for this processor.");
        }
        System.out.println(x: "--- [TEMPLATE END] ---\n");
    }

    private boolean checkFileExists(File file) {
        return file.exists() && file.isFile();
    }

    private void scanForViruses(File file) {
        System.out.println(">> Security: Scanning " + file.getName() + " for viruses... OK.");
    }

    private void sendNotification(String user, String filename) {
        System.out.println(">> Notification: Informing admin that " + user + " uploaded " + filename);
    }
}
```

ImageProcessor це реалізація алгоритму для зображень. Виступає як Concrete Class.

```
1 package com.example.ftp.service.template;
2
3 import org.springframework.stereotype.Component;
4 import java.io.File;
5
6 @Component
7 public class ImageProcessor extends FileProcessingTemplate {
8
9     @Override
10    protected boolean isValidType(File file) {
11        return file.getName().endsWith(suffix: ".jpg") || file.getName().endsWith(suffix: ".png");
12    }
13
14    @Override
15    protected void extractMetadata(File file) {
16        System.out.println(">> Image Processor: Analyzing dimensions of " + file.getName());
17        System.out.println(x: "    -> Result: Resolution 1920x1080.");
18    }
19 }
```

TextDocProcessor це реалізація алгоритму спеціально для текстових документів. Виступає як Concrete Class.

```
1 package com.example.ftp.service.template;
2
3 import org.springframework.stereotype.Component;
4 import java.io.File;
5
6 @Component
7 public class TextDocProcessor extends FileProcessingTemplate {
8
9     @Override
10    protected boolean isValidType(File file) {
11        return file.getName().endsWith(suffix: ".txt") || file.getName().endsWith(suffix: ".log");
12    }
13
14    @Override
15    protected void extractMetadata(File file) {
16        System.out.println(">> Text Processor: Counting lines and words in " + file.getName());
17        System.out.println(x: "    -> Result: 150 lines found (Simulated).");
18    }
19
20    @Override
21    protected boolean shouldCreateBackup() {
22        return true;
23    }
24 }
```

Результат:

```
--- [TEMPLATE START] Processing test.txt ---
>> Security: Scanning test.txt for viruses... OK.
>> Text Processor: Counting lines and words in test.txt
    -> Result: 150 lines found (Simulated).
>> Backup: Creating backup copy of test.txt to /backup folder.
>> Notification: Informing admin that admin uploaded test.txt
--- [TEMPLATE END] ---
```

```
--- [TEMPLATE START] Processing Зн?мок екрана 2025-12-01 085210.png ---
>> Security: Scanning Зн?мок екрана 2025-12-01 085210.png for viruses... OK.
>> Image Processor: Analyzing dimensions of Зн?мок екрана 2025-12-01 085210.png
    -> Result: Resolution 1920x1080.
>> Notification: Informing admin that admin uploaded Зн?мок екрана 2025-12-01 085210.png
--- [TEMPLATE END] ---
```

На знімках показано результат роботи обробки файлів з розширенням .txt та .png через термінал, також патерн виконує поверхневу перевірку файлу та повідомляє адміністратора про передачу файлів.

Висновки:

Я вивчив структуру шаблонів «Mediator», «Facade», «Bridge», «Template method» та навчився застосовувати їх в реалізації програмної системи.