

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 5
з дисципліни «Технології розроблення
програмного забезпечення»
Тема: «Патерни проектування»

Виконав:
студент групи IA-32
Задорожний Костянтин
Леонідович

Перевірив:
Мягкий Михайло
Юрійович

Київ - 2025

Тема: Патерни проектування.

Мета: Вивчити структуру шаблонів «Adapter», «Builder», «Command», «Chain of responsibility», «Prototype» 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 повинен вміти коректно обробляти і відправляти відповіді під протоколу FTP, з можливістю створення користувачів (з паролями) 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 «Ланцюжок відповідальності» (Chain of Responsibility).

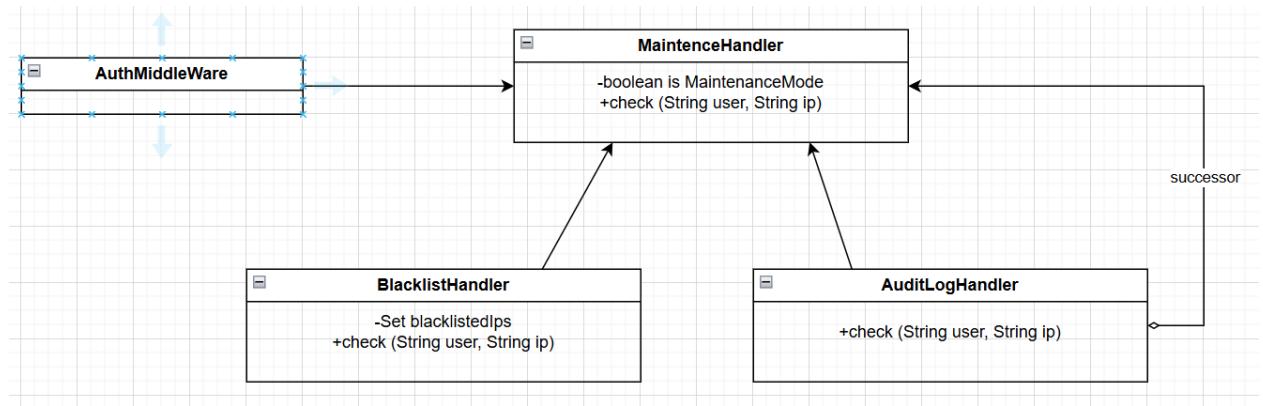
У FTP-сервері цей патерн дозволить створити ланцюжок перевірок перед тим, як користувач отримає доступ (автентифікацію). Спочатку перевіряємо, чи не ведуться технічні роботи, потім — чи не заблокована IP-адреса, і лише потім переходимо до перевірки пароля.

З теоретичних відомостей:

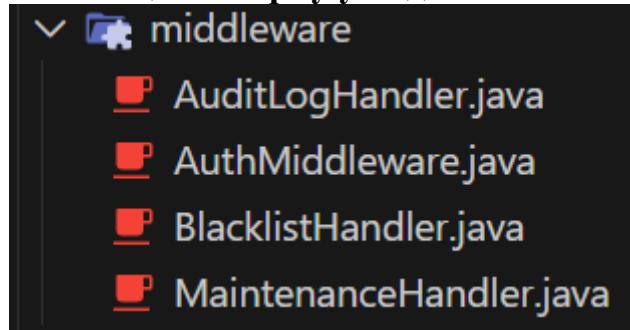
Переваги та недоліки:

- + Зменшує залежність між клієнтом та обробниками: клієнт не знає хто обробить запит, а обробники не знають структуру ланцюжка.
- + Реалізовує додаткову гнучкість в обробці запиту: легко добавити або вилучити з ланцюжка нові обробники.
- Запит може залишитися ніким не опрацьованим: запит не має вказаного обробника, тому може бути не опрацьованим.

Діаграма структури патерну Chain of Responsibility для FTP server:



Реалізація патерну у коді:



AuthMiddleware.java:

```
1 package com.example.ftp.service.middleware;
2 import org.slf4j.Logger;
3 import org.slf4j.LoggerFactory;
4
5 public abstract class AuthMiddleware {
6     private AuthMiddleware next;
7     protected static final Logger logger = LoggerFactory.getLogger(AuthMiddleware.class);
8
9     public AuthMiddleware linkWith(AuthMiddleware next) {
10         this.next = next;
11         return next;
12     }
13
14     public abstract boolean check(String username, String userIp);
15
16     protected boolean checkNext(String username, String userIp) {
17         if (next == null) return true;
18         return next.check(username, userIp);
19     }
20 }
```

BlacklistHandler.java:

```
1 package com.example.ftp.service.middleware;
2
3 import org.springframework.stereotype.Component;
4 import java.util.Set;
5
6 @Component
7 public class BlacklistHandler extends AuthMiddleware {
8     private final Set<String> blacklistedIps = Set.of("192.168.1.666");
9
10    @Override
11    public boolean check(String username, String userIp) {
12        if (blacklistedIps.contains(userIp)) {
13            logger.error("Login blocked: IP " + userIp + " is blacklisted.");
14            return false;
15        }
16        return checkNext(username, userIp);
17    }
18 }
```

MaintenanceHandler.java:

```

1 package com.example.ftp.service.middleware;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class MaintenanceHandler extends AuthMiddleware {
7     private boolean isMaintenanceMode = false;
8
9     @Override
10    public boolean check(String username, String userIp) {
11        if (isMaintenanceMode) {
12            if ("admin".equals(username))
13                return checkNext(username, userIp);
14            logger.warn("Login denied: Maintenance mode. User: " + username);
15            return false;
16        }
17        return checkNext(username, userIp);
18    }
19 }

```

AuditLogHandler.java:

```

1 package com.example.ftp.service.middleware;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class AuditLogHandler extends AuthMiddleware {
7     @Override
8     public boolean check(String username, String userIp) {
9         logger.info("[AUDIT] Auth attempt: User={} IP={}", username, userIp);
10        return checkNext(username, userIp);
11    }
12 }

```

Щоб перевірити роботу класів безпеки Chain of Responsibility необхідно:

- Додати локальну айпі адресу у чорний список (BlacklistHandler.java)
`private final Set<String> blacklistedIps = Set.of("192.168.1.666", "127.0.0.1");`
- Змінити значення булевої функції isMaintenanceMode для початку технічної зупинки (MaintenanceHandler.java).

Відповідні результати:

```

INFO 12860 --- [pool-4-thread-1] o.a.f.listener.nio.FtpLoggingFilter      : SENT: 331 User name okay, need password for user1.

```

```

INFO 12860 --- [pool-4-thread-1] o.a.f.listener.nio.FtpLoggingFilter      : RECEIVED: PASS *****

```

```

ERROR 12860 --- [pool-4-thread-1] c.e.f.service.middleware.AuthMiddleware : Login blocked: IP 127.0.0.1 is blacklisted.

```

```

WARN 12860 --- [pool-4-thread-1] org.apache.ftpserver.command.impl.PASS   : User failed to log in

```

```

WARN 12860 --- [pool-4-thread-1] org.apache.ftpserver.command.impl.PASS   : Login failure - user1

```

```

INFO 12860 --- [pool-4-thread-1] o.a.f.listener.nio.FtpLoggingFilter      : SENT: 530 Authentication failed.

```

```

INFO 12860 --- [pool-4-thread-1] o.a.f.listener.nio.FtpLoggingFilter      : CLOSED

```

```

INFO 3560 --- [pool-4-thread-2] o.a.f.listener.nio.FtpLoggingFilter      : SENT: 331 User name okay, need password for user1.

```

```

INFO 3560 --- [pool-4-thread-2] o.a.f.listener.nio.FtpLoggingFilter      : RECEIVED: PASS *****

```

```

WARN 3560 --- [pool-4-thread-2] c.e.f.service.middleware.AuthMiddleware : Login denied: Maintenance mode. User: user1

```

```

WARN 3560 --- [pool-4-thread-2] org.apache.ftpserver.command.impl.PASS   : User failed to log in

```

```

WARN 3560 --- [pool-4-thread-2] org.apache.ftpserver.command.impl.PASS   : Login failure - user1

```

```

INFO 3560 --- [pool-4-thread-2] o.a.f.listener.nio.FtpLoggingFilter      : SENT: 530 Authentication failed.

```

```

INFO 3560 --- [pool-4-thread-2] o.a.f.listener.nio.FtpLoggingFilter      : CLOSED

```

Висновки:

Я вивчив структуру шаблонів «Adapter», «Builder», «Command», «Chain of responsibility», «Prototype» та навчився застосовувати їх в реалізації програмної системи.