

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

Лабораторна робота № 4
з дисципліни «Технології розроблення
програмного забезпечення»
Тема: «Вступ до паттернів проектування»

Виконав:
студент групи IA-32
Задорожний Костянтин
Леонідович

Перевірив:
Мягкий Михайло
Юрійович

Київ - 2025

Тема: Вступ до паттернів проектування.

Мета: Вивчити структуру шаблонів «Singleton», «Iterator», «Proxy», «State», «Strategy» 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-сервер 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 використав патерн «Стратегія» (Strategy).

Згідно з завданням предметної області, FTP-сервер повинен мати функцію «обмеження максимальної швидкості поширення глобально і окремо для кожного облікового запису». Патерн Strategy ідеально підходить для цього,

адже він дозволить динамічно змінювати алгоритм розрахунку швидкості для користувача без зміни основного коду менеджера користувачів.

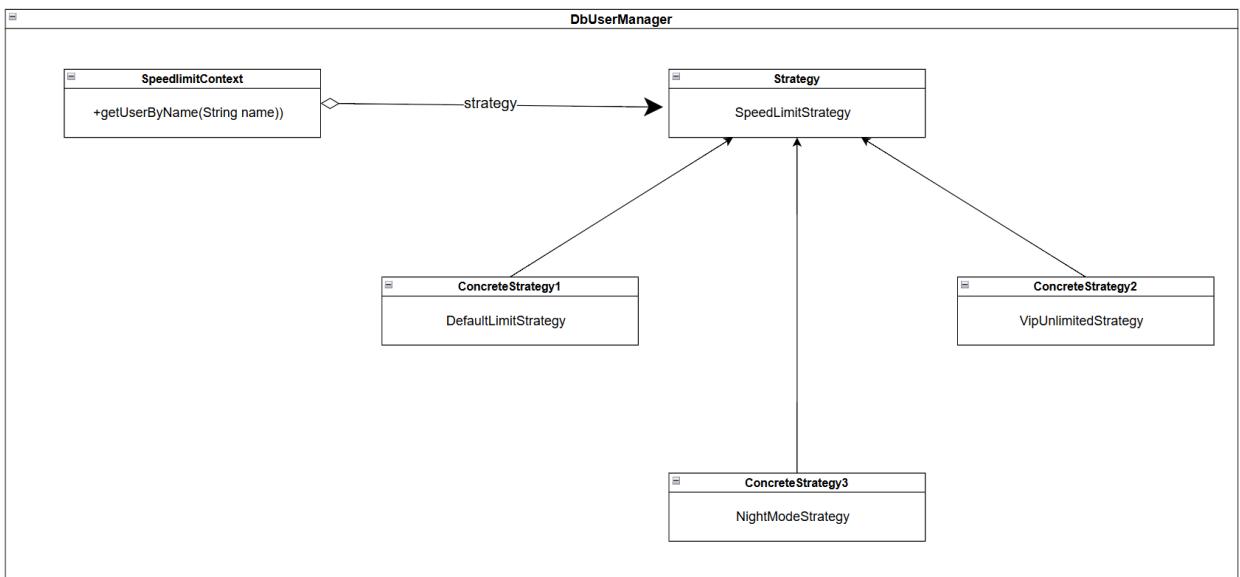
З теоретичних відомостей:

Патерн «Стратегія» (Strategy) — дозволяє змінювати деякий алгоритм поведінки об'єкта іншим алгоритмом, що досягає ту ж мету іншим способом.

Контекст: Зберігає посилання на об'єкт Стратегії і спілкується з ним через інтерфейс.

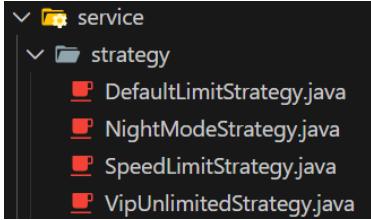
Стратегія: Спільний інтерфейс для всіх алгоритмів.

Діаграма структури патерну Strategy для FTP server:



- **SpeedLimitContext** виступає як **Context**. Він вирішує, яку стратегію обрати.
- **SpeedLimitStrategy** є спільним інтерфейсом **Strategy**.
- Класи **DefaultLimitStrategy**, **VipUnlimitedStrategy** — це **ConcreteStrategies**, що містять специфічну логіку розрахунку.

Реалізація патерну у коді:



DefaultLimitStrategy:

```
1 package com.example.ftp.service.strategy;
2
3 import com.example.ftp.model.FtpUser;
4 import org.apache.ftpserver.usermanager.impl.TransferRatePermission;
5 import org.springframework.stereotype.Component;
6
7 @Component("defaultStrategy")
8 public class DefaultLimitStrategy implements SpeedLimitStrategy {
9     @Override
10    public TransferRatePermission calculateLimit(FtpUser user) {
11        return new TransferRatePermission(user.getMaxDownloadRate(), user.getMaxUploadRate());
12    }
13}
14}
```

NightModeStrategy:

```
1 package com.example.ftp.service.strategy;
2
3 import com.example.ftp.model.FtpUser;
4 import org.apache.ftpserver.usermanager.impl.TransferRatePermission;
5 import org.springframework.stereotype.Component;
6 import java.time.LocalTime;
7
8 @Component("nightStrategy")
9 public class NightModeStrategy implements SpeedLimitStrategy {
10    @Override
11    public TransferRatePermission calculateLimit(FtpUser user) {
12        LocalTime now = LocalTime.now();
13        // Якщо зараз між 00:00 та 08:00 – бе зліміт
14        if (now.isAfter(LocalTime.MIDNIGHT) && now.isBefore(LocalTime.of(hour: 8, minute: 0))) {
15            return new TransferRatePermission(maxDownloadRate: 0, maxUploadRate: 0);
16        }
17        // Вдень використовуємо стандартні обмеження користувача
18        return new TransferRatePermission(user.getMaxDownloadRate(), user.getMaxUploadRate());
19    }
20}
```

SpeedLimitStrategy:

```
1 package com.example.ftp.service.strategy;
2
3 import com.example.ftp.model.FtpUser;
4 import org.apache.ftpserver.usermanager.impl.TransferRatePermission;
5
6 public interface SpeedLimitStrategy {
7     TransferRatePermission calculateLimit(FtpUser user);
8 }
```

VipUnlimitedStrategy:

```

1 package com.example.ftp.service.strategy;
2
3 import com.example.ftp.model.FtpUser;
4 import org.apache.ftpserver.usermanager.impl.TransferRatePermission;
5 import org.springframework.stereotype.Component;
6
7 @Component("vipStrategy")
8 public class VipUnlimitedStrategy implements SpeedLimitStrategy {
9     @Override
10    public TransferRatePermission calculateLimit(FtpUser user) {
11        // 0 означає "безліміт" в Apache FtpServer
12        return new TransferRatePermission(maxDownloadRate: 0, maxUploadRate: 0);
13    }
14 }

```

Для перевірки роботи класів патерну Strategy необхідно створити трьох користувачів:

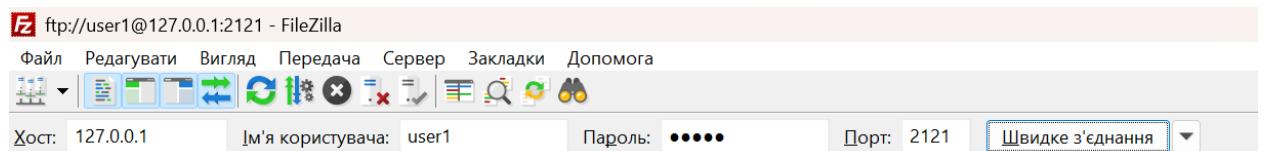
admin – створений за замовчуванням в базі даних, має необмежену швидкість передачі даних;

vipClient – надається клієнту якщо в імені присутній суфікс “_vip”, також надає безліміт;

nightClient – надається клієнту якщо в імені присутній суфікс “_night”, надає безліміт тільки в нічний час відносно часового поясу сервера.

Перевірити точну швидкість передачі немає можливості. В застосунку FileZilla при підключені проекту та передачі файлів на віддалений сайт можна відчути різницю у затраченому часі на передачу одного і того самого файлу, якщо робити це з привілеями адміна та простого користувача.

Підключення проекту до FileZilla:



Кількість переданої інформації користувачем за 9 хвилин:

2 файли. Загальний розмір: 2 711 598 байт

Роблячи нескладні розрахунки отримаємо швидкість 5021 байт/секунду у користувача, що співпадає з налаштуваннями користувача:

user1	C:/ftp/user3	true	5000
-------	--------------	------	------

Кількість переданої інформації адміном за кілька секунд:

2 файли. Загальний розмір: 7 431 904 байти

Бачимо, що безлімітна передача даних проходить майже моментально. Таким чином я переконався у справній роботі коду обмеження швидкості передачі даних для різних типів користувачів.

Висновок:

Я вивчив структуру шаблонів «Singleton», «Iterator», «Proxy», «State», «Strategy» та навчився застосовувати їх в реалізації програмної системи.