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 ІА-34
Сухоручкін Гліб

Перевірив:
Асистент кафедри ІСТ
Мягкий М.Ю.

Київ – 2025

Зміст:

Вступ

Метою цієї роботи є вивчити структуру шаблонів «Singleton», «Iterator», «Proxy», «State», «Strategy» та навчитися застосовувати їх в реалізації програмної системи.

Проєкторваною системою цієї лабороторної роботи є FTP-server (використовуючи state, builder, memento, template method, visitor, client-server).

FTP-сервер повинен вміти коректно обробляти і відправляти відповіді по 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.

Теоретичні відомості

Будь-який патерн проєктування, використовуваний при розробці інформаційних систем, являє собою формалізований опис, який часто зустрічається в завданнях проєктування, вдале рішення даної задачі, а також рекомендації по застосуванню цього рішення в різних ситуаціях. Крім того, патерн проєктування обов'язково має загальновживане найменування. Правильно сформульований патерн проєктування дозволяє, відшукавши одного разу вдале рішення, користуватися ним знову і знову. Варто підкреслити, що важливим початковим етапом при роботі з патернами є адекватне моделювання розглянутої предметної області. Це є необхідним як для отримання належним чином формалізованої постановки задачі, так і для вибору відповідних патернів проєктування.

Відповідне використання патернів проєктування дає розробнику ряд незаперечних переваг. Наведемо деякі з них. Модель системи, побудована в межах патернів проєктування, фактично є структурованим виокремленням тих елементів і зв'язків, які значимі при вирішенні поставленого завдання. Крім цього, модель, побудована з використанням патернів проєктування, більш проста і наочна у вивченні, ніж стандартна модель. Проте, не дивлячись на простоту і наочність, вона дозволяє глибоко і всебічно опрацювати архітектуру розроблюваної системи з використанням спеціальної мови.

Застосування патернів проєктування підвищує стійкість системи до зміни вимог та спрощує неминуче подальше доопрацювання системи. Крім того, важко переоцінити роль використання патернів при інтеграції інформаційних систем організації. Також слід зазначити, що сукупність патернів проєктування, по суті, являє собою єдиний словник проєктування, який, будучи уніфікованим засобом, незамінний для спілкування розробників один одним.

Таким чином шаблони представляють собою, підтверджені роками розробок в різних компаніях і на різних проєктах, «ескізи» архітектурних рішень, які зручно застосовувати у відповідних обставинах.

Хід роботи

Діаграма класів

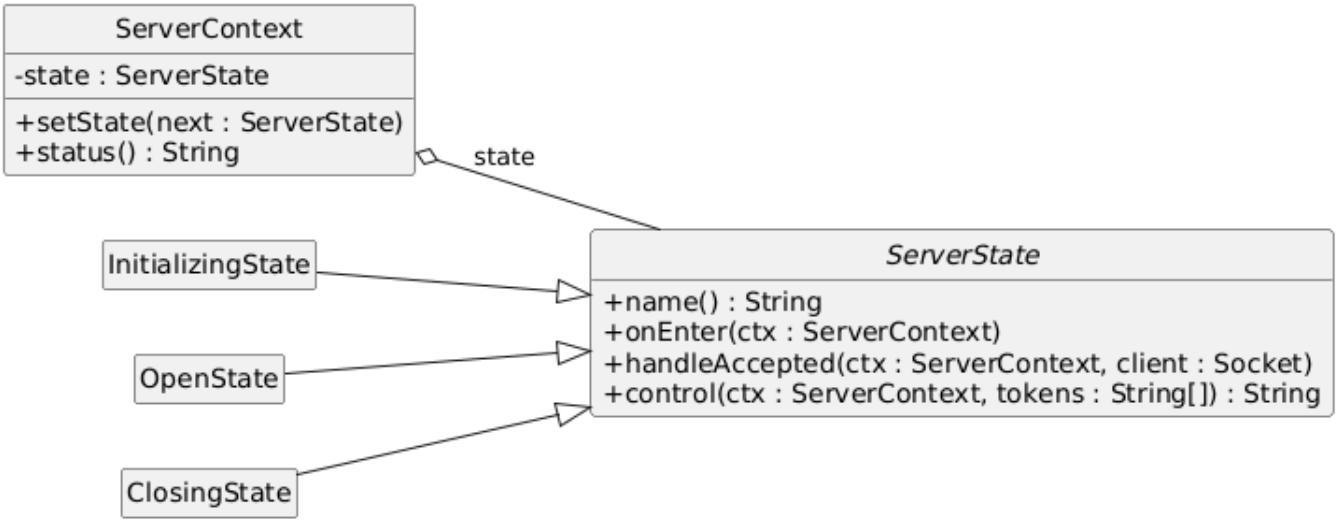


Рис 1. Патерн «Стан» у FTP-сервері.

Пояснення діаграми классів

ServerContext – «контекст» сервера, який має поле state: ServerState і керує перемиканням станів через setState(next: ServerState); метод status(): String повертає текстове представлення поточного стану.

ServerState – абстрактний інтерфейс/базовий клас стану. Визначає контракт:

name(): String – назва стану;

onEnter(ctx: ServerContext) – хук, який викликається під час входу в стан;

handleAccepted(ctx: ServerContext, client: Socket) – обробка нового прийнятого з’єднання;

control(ctx: ServerContext, tokens: String[]): String – обробка керуючих команд і повернення рядкової відповіді.

InitializingState, OpenState, ClosingState – конкретні реалізації ServerState, кожна зі своєю поведінкою для описаних методів.

Зв’язки: ServerContext утримує посилання на поточний ServerState (поле state), а три класи станів наслідують/реалізують ServerState. Стан може ініціювати перехід, викликаючи ctx.setState(…).

Вихідний код

https://github.com/GlebTiKsTRPZ/lab4

Відповіді на питання до лабораторної роботи

1. Що таке шаблон проєктування?

Формалізований опис повторюваної в проєктуванні задачі, перевірене рішення та рекомендації щодо його застосування; патерн має загальновживану назву і дозволяє багаторазово використовувати знайдене вдале рішення.

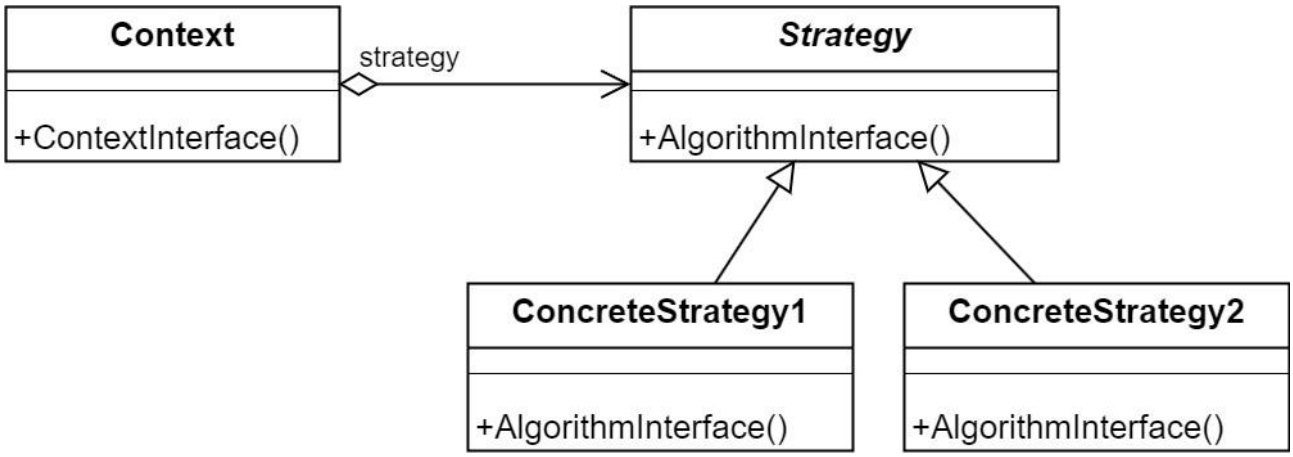
2. Навіщо використовувати шаблони проєктування?

Вони структурують модель системи навколо суттєвих елементів/зв’язків, полегшують вибір рішень під час моделювання та дають переваги на кшталт повторного використання перевірених підходів.

3. Яке призначення шаблону «Стратегія»?

Дозволяє взаємозамінно підмінювати алгоритм поведінки об’єкта іншим алгоритмом, що досягає тієї ж мети; зручно, коли існують різні «політики» обробки даних.

4. Нарисуйте структуру шаблону «Стратегія».



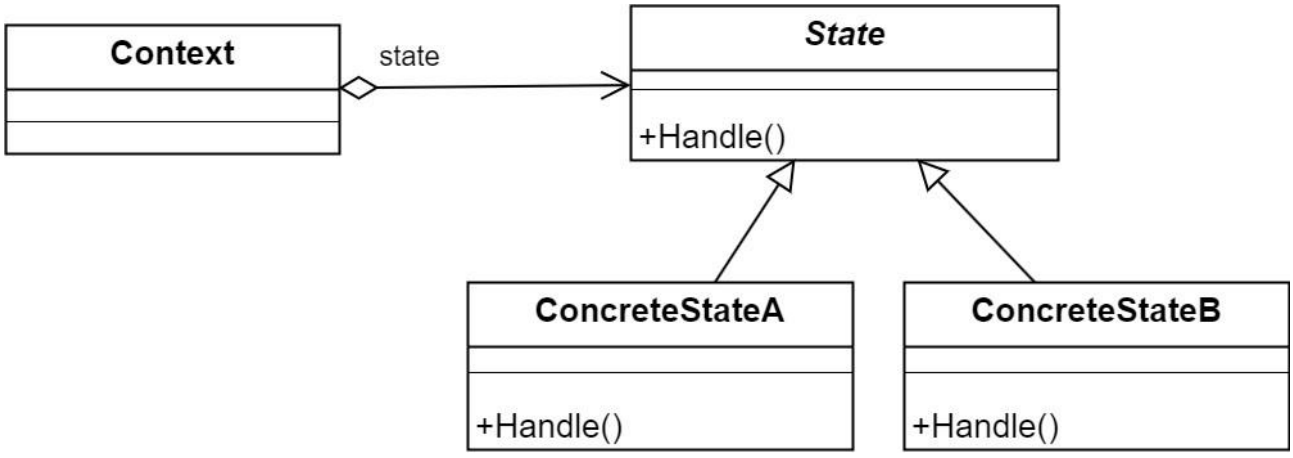
5. Які класи входять в шаблон «Стратегія», та яка між ними взаємодія?

Є контекст (Context), інтерфейс стратегії та конкретні стратегії; контекст тримає посилання на стратегію і під час роботи може її замінювати.

6. Яке призначення шаблону «Стан»?

Відповідь: Змінювати логіку роботи об’єкта залежно від його внутрішнього стану, інкапсулюючи поведінку у класи станів.

7. Нарисуйте структуру шаблону «Стан».



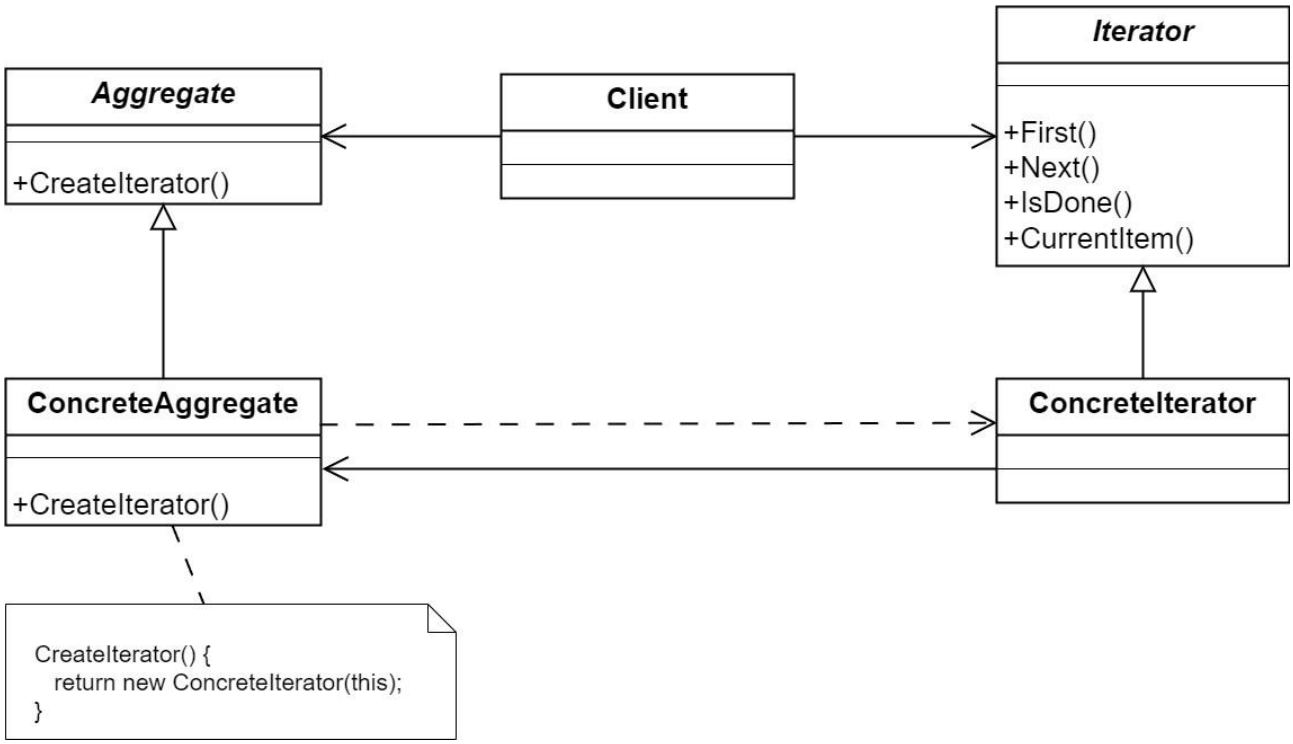
8. Які класи входять в шаблон «Стан», та яка між ними взаємодія?

Контекст і інтерфейс стану з конкретними станами (напр., OpenState, ClosingState); контекст делегує поведінку активному стану та перемикає стани через визначені переходи.

9. Яке призначення шаблону «Ітератор»?

Надає спосіб послідовно обходити елементи колекції без розкриття її внутрішнього подання, розділяючи зберігання даних і перебір.

10. Нарисуйте структуру шаблону «Ітератор».



11. Які класи входять в шаблон «Ітератор», та яка між ними взаємодія?

Колекція (агрегат) та ітератор: ітератор надає операції First/Next/IsDone/CurrentItem і проходить елементи незалежно від реалізації колекції.

12. В чому полягає ідея шаблона «Одинак»?

Гарантувати існування рівно одного екземпляра класу та надати до нього глобальну точку доступу (типові мотиви — єдині налаштування, унікальний сеанс тощо).

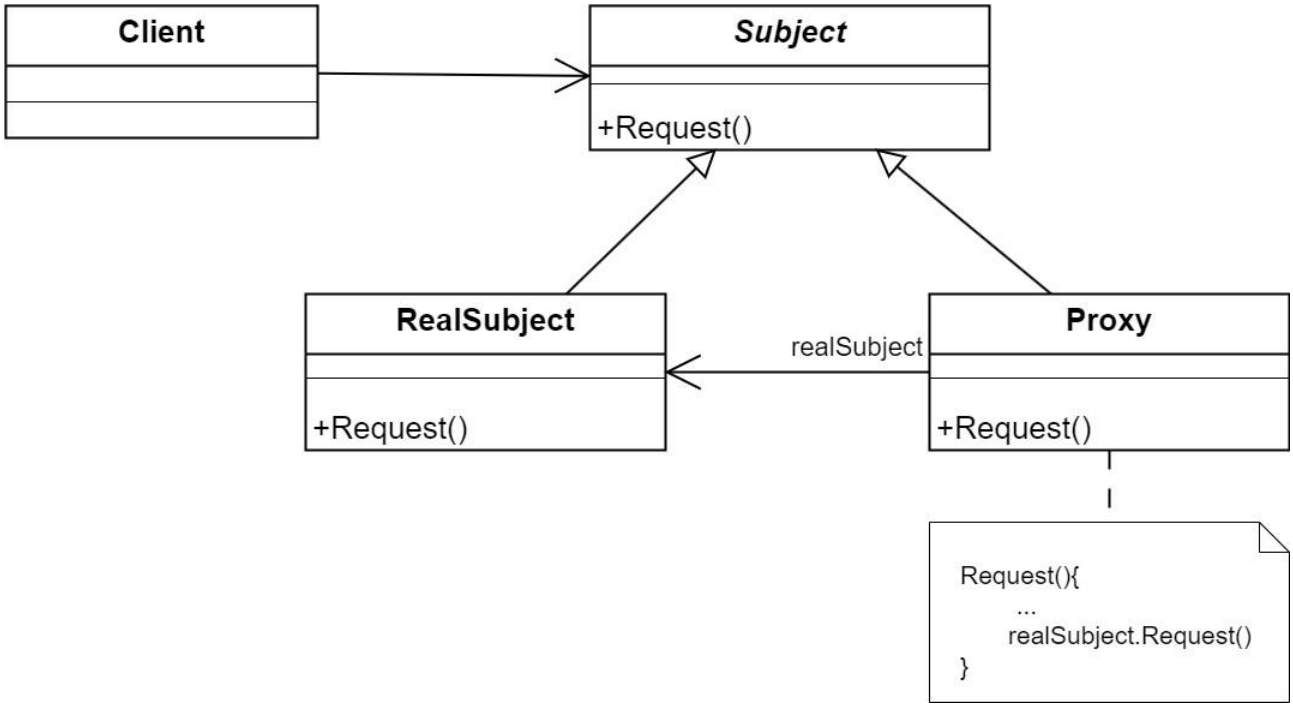
13. Чому шаблон «Одинак» вважають «анти-шаблоном»?

Через глобальний стан і порушення принципу єдиної відповідальності — такі об’єкти важко тестувати й підтримувати; патерн часто маскує слабкий дизайн.

14. Яке призначення шаблону «Проксі»?

Додати проміжний об’єкт-заступник, що спрощує доступ або розширює функціональність реального об’єкта, зберігаючи той самий інтерфейс для клієнта.

15. Нарисуйте структуру шаблону «Проксі».



16. Які класи входять в шаблон «Проксі», та яка між ними взаємодія?

Інтерфейс сервісу (напр., IDocSignManager), реальна реалізація сервісу та проксі (DocSignManagerProxy), з яким працює клієнт; проксі делегує виклики реальному об’єкту й може додавати кешування/контроль звернень.

Висновок

У цій лабораторній роботі впроваджено патерн «Стан» для FTP-сервера (Initializing -> Open -> Closing), що централізував керування в ServerContext і спростив обробку підключень та команд.