

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 ІА-32
Задорожний Костянтин
Леонідович

Перевірів:
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 (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 вирішив використати 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 додати 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:

