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 «Технологія розробки програмного забезпечення»

Тема: «ШАБЛОНИ «SINGLETON», «ITERATOR», «PROXY», «STATE», «STRATEGY»»

Виконав:
студент 3 курсу,
групи ІА-12
Одемчук Н. О.

Перевірив:
ас. Колеснік В.М.

Київ 2023

Тема: ШАБЛОНИ «SINGLETON», «ITERATOR», «PROXY», «STATE», «STRATEGY»

Мета: Застосування одного з розглянутих шаблонів при реалізації програми.

Завдання:

Клієнт для IRC-чатів з можливістю вказівки порту і адреси з'єднання, підтримка базових команд (підключення до чату, створення чату, установка імені, реєстрація, допомога і т.д.), отримання метаданих про канал.

Хід роботи

Шаблон стратегії – це патерн проектування поведінки, який визначає сім'ю алгоритмів, інкапсулює кожен алгоритм і робить їх взаємозамінними. Це дозволяє клієнту вибирати відповідний алгоритм під час виконання програми. У контексті мого коду клієнтський код відповідає за виконання різних команд IRC, і я хочу зробити ці команди взаємозамінними без зміни самого класу MessageHandler. Ось як шаблон стратегії застосовується в моєму коді:

Інтерфейс стратегії (Strategy): У цьому випадку кожен клас стратегії (наприклад, JoinChannelStrategy, LeaveChannelStrategy) має спільний метод execute і наслідує інтерфейс стратегії IRCCommandStrategy. Цей метод інкапсулює конкретну поведінку, пов'язану з кожною командою IRC.

Конкретні стратегії (JoinChannelStrategy, LeaveChannelStrategy і т. д.): Кожен конкретний клас стратегії реалізує метод execute, надаючи конкретну поведінку для певної команди IRC.

Контекст (MessageHandler): Клас MessageHandler – це контекст, в якому

використовуються стратегії. Він має посилання на поточну стратегію і використовує її для виконання конкретної команди IRC.

Клієнт (Використання MessageHandler): У методі command_choice класу MessageHandler вибирається відповідна стратегія на основі отриманої команди IRC. Потім викликається метод execute стратегії, що забезпечує чистий і модульний спосіб обробки різних команд.

Розглянемо ключові компоненти:

Гнучкість: Нові команди можна додавати, створюючи нові класи стратегій без модифікації існуючого класу MessageHandler. Кожна команда інкапсулює свою поведінку у власному класі стратегії.

Читабельність: Код стає більш зрозумілим і піддається технічному обслуговуванню, оскільки логіка кожної команди міститься у власному класі стратегії. Якщо вам потрібно розуміти чи змінити поведінку конкретної команди, вам достатньо переглянути відповідний клас стратегії.

Масштабованість: З ростом кількості команд шаблон стратегії дозволяє ефективніше управляти і організовувати код. Додавання нових стратегій без модифікації існуючих дозволяє легко розширювати функціонал.

У підсумку, шаблон стратегії у моєму коді дозволяє створити більш модульний і розширюваний дизайн, спрощуючи додавання, модифікацію чи обслуговування різних команд IRC незалежно одна від одної.

Висновки: У цій лабораторній роботі я реалізував шаблон «STRATEGY».