

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 «Технологія розробки програмного забезпечення»

Тема: «ШАБЛони «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF
RESPONSIBILITY», «PROTOTYPE»»

Виконав:

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

Перевірів:

ас. Колеснік В.М.

Тема: ШАБЛОНИ «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF RESPONSIBILITY», «PROTOTYPE»

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

В розробці програмного забезпечення, паттерн Команда є паттерном проектування поведінки, який перетворює запит в самостійний об'єкт. Цей об'єкт містить всю інформацію про запит, дозволяючи параметризацію клієнтів різними запитами, утримання запитів в черзі та реєстрацію операцій. Ключові компоненти патерну Команда:

1. Команда (Інтерфейс і Абстрактний Клас):

- Декларує інтерфейс і абстрактний клас з методом `execute`, який конкретні команди повинні реалізовувати.
- Представляє команду в системі.

2. Конкретна Команда:

- Реалізує інтерфейс `Command` або розширює абстрактний клас.
- Визначає зв'язок між конкретною дією та об'єктом, що викликає цю дію.
- Зберігає додаткові параметри, необхідні для виконання дії.

3. Інвокер:

- Запитує команду виконати операцію.
- Утримує посилання на об'єкт команди, але не знає конкретних деталей команди.

4. Отримувач:

- Знає, як виконати операцію, пов'язану з певною командою.

- Викликається Конкретною Командою для виконання запиту.

5. Клієнт:

- Створює та конфігурує об'єкти команд.
- Асоціює команду з отримувачем.

Тепер давайте розглянемо ці компоненти у моєму коді:

- Команда (Абстрактний Клас): Клас `Command` є абстрактним базовим класом для класів команд IRC. Він декларує метод `execute`.
- Конкретна Команда:
 - Класи `JoinChannelCommand`, `LeaveChannelCommand`, `ListUsersCommand`, `ChannelInfoCommand`, `ChangeNicknameCommand`, `DisplayHelpCommand`, та `QuitCommand` є конкретними класами команд. Кожен реалізує метод `execute`, визначаючи конкретні дії, пов'язані з командами IRC.
- Інвокер:
 - Клас `MessageHandler` виступає в ролі інвокера. Він вибирає та виконує команди IRC на основі введення користувача.
- Отримувач:
 - Клас `ServerHandler` виступає в ролі отримувача. Він виконує дії, пов'язані з командами IRC, такі як підключення, відключення та відправлення даних на сервер.
- Клієнт:
 - Клас `IRCCClient` виступає в ролі клієнта. Він створює та конфігурує об'єкти команд (`Command`), враховуючи введення користувача.

Паттерн Команда в моєму коді дозволяє інкапсулювати різні команди IRC у вигляді об'єктів, що полегшує розширення та додавання нових команд без змінення існуючого коду. Він також сприяє розділенню відправника (Інвокера) та отримувача, сприяючи гнучкості та зручності у підтримці кодової бази.

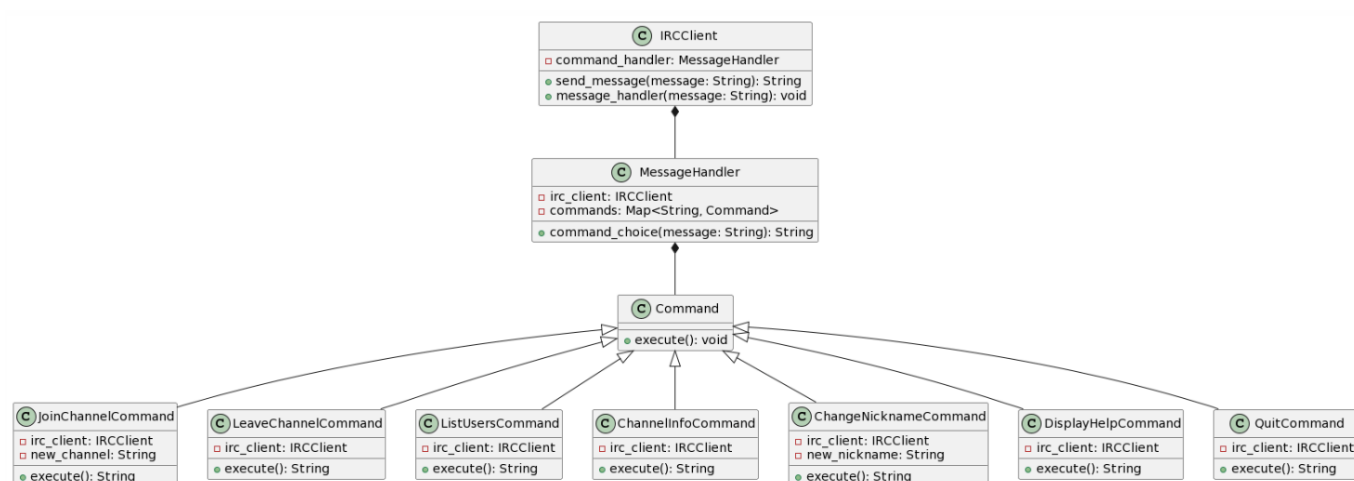


Рисунок 5.1 – Діаграма шаблону «COMMAND»

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