

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: «»

Виконав:

студент групи ІА-34

Сухоручкін Гліб

Перевірив:

Асистент кафедри ІСТ

Мягкий М.Ю.

Зміст:

1. Вступ
2. Теоретичні відомості
3. Хід роботи
4. Діаграма класів
5. Вихідний код
6. Висновки

Вступ

Метою цієї роботи є вивчити структуру шаблонів «Adapter», «Builder», «Command», «Chain of responsibility», «Prototype» та навчитися застосовувати їх в реалізації програмної системи.

Проекторваною системою цієї лабораторної 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

Шаблон «Builder»

Призначення патерну: Шаблон «Builder» (Будівельник) використовується для відділення процесу створення об'єкту від його представлення [6]. Це доречно у випадках, коли об'єкт має складний процес створення (наприклад, Web- сторінка як елемент повної відповіді web- сервера) або коли об'єкт повинен мати декілька різних форм створення (наприклад, при конвертації тексту з формату у формат).

Проблема: Візьмемо процес побудови відповіді на запит web-сервера. Побудова складається з наступних частин: додавання стандартних заголовків (дата/час, ім'я сервера, інш.), код статусу (після пошуку відповідної сторінки на сервері), заголовки відповіді (тип вмісту, інш.), утримуване, інше.

Рішення: Кожен з цих етапів може бути абстрагований в окремий метод будівельника. Це дасть наступні вигоди:

- Гнучкіший контроль над процесом створення сторінки;
- Незалежність від внутрішніх змін – наприклад, зміна назви сервера не сильно порушить процес побудови відповіді;

Переваги та недоліки:

+ Дозволяє використовувати один і той самий код для створення різноманітних продуктів.

- Клієнт буде прив'язаний до конкретних класів будівельників, тому що в інтерфейсі будівельника може не бути методу отримання результату.

Хід роботи

Діаграма класів

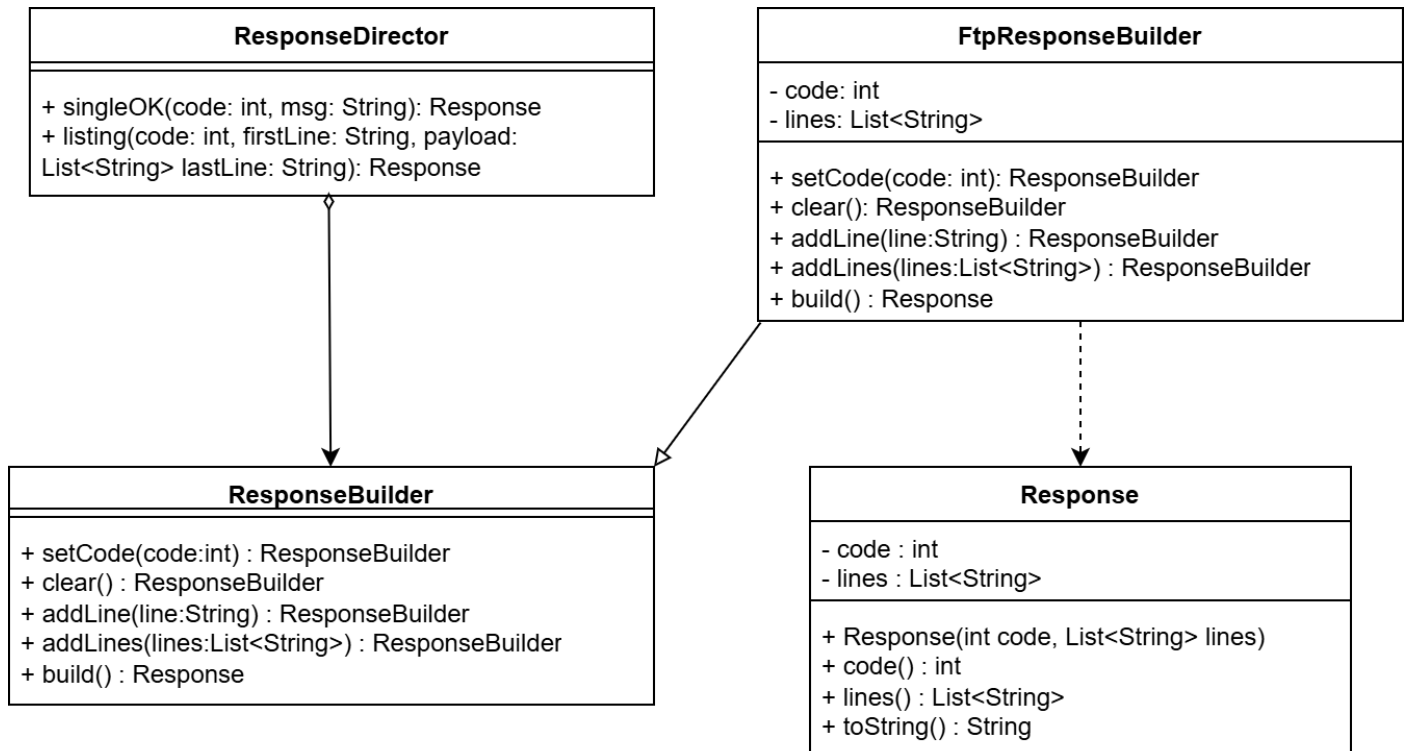


Рис. 1 – Патерн «Builder» у FTP-сервері

Пояснення діаграми класів

Response – «продукт» із кодом відповіді та списком рядків; сам відповідає за форматування FTP-відповідей у `toString()`.

ResponseBuilder – інтерфейс будівельника з fluent-методами для коду та рядків.

FtpResponseBuilder – конкретний будівельник, який накопичує стан (код + рядки) і створює **Response** у `build()`.

ResponseDirector – утилітний «директор», що інкапсулює готові сценарії побудови: `singleOK(...)` для однотипних відповідей і `listing(...)` для багаторядкових (XYZ-.... ... XYZ ...).

Вихідний код

<https://github.com/GlebTiKsTRPZ/lab5>

Висновок

У цій лабораторній роботі впроваджено патерн «Будівельник» для формування FTP-відповідей: створено Response, ResponseBuilder/FtpResponseBuilder та ResponseDirector, що відокремлюють процес побудови від представлення. Це уніфікувало одно- й багаторядкові відповіді (зокрема PWD і LIST) та спростило підтримку й подальше розширення логіки сервера.