МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Навчально-науковий інститут комп’ютерного моделювання, прикладної фізики та математики

Звіт

З лабораторної роботи №6

З дисципліни:“ Ефективність та якість архітектурних рішень інформаційних систем ”

На тему:

«Патерн проєктування Замісник (Proxy) »

Виконав:

студент групи ІКМ-М225В

Загорулько Віталій Олегович

Харків 2025

Мета роботи

Метою даної лабораторної роботи є ґрунтовне вивчення та практичне засвоєння структурного патерна проєктування **Замісник** (Proxy), а також набуття навичок його застосування для надання контрольованого доступу до іншого об'єкта, зокрема, для додавання допоміжної функціональності (кешування) без внесення змін до початкового класу.

Завдання

– У межах виконання лабораторної роботи необхідно виконати такі завдання:

– Ознайомитися з теоретичними засадами та основними ідеями патерна проєктування Замісник.

– Створити інтерфейс Downloader та його початкову реалізацію SimpleDownloader.

– Додати до системи механізм кешування завантажених даних, використовуючи патерн Замісник, при цьому не модифікуючи клас SimpleDownloader.

– Створити структуру класів та методів, що вирішує описане завдання, та приклад клієнтського коду, що демонструє роботу кешування.

Теоретичні відомості та Опис Проєкту

Патерн Замісник (Proxy) — це структурний патерн проєктування, який створює об'єкт-замінник або "заповнювач" (placeholder) для іншого об'єкта, щоб контролювати доступ до нього та додавати додаткову логіку перед або після його виклику . Ключова вимога патерна полягає в тому, що як оригінальний об'єкт (Сервіс), так і його Замісник мають реалізовувати один і той самий інтерфейс, забезпечуючи прозорість для клієнтського коду. Замісники можуть бути використані для захисту, відкладеної ініціалізації, моніторингу або, як у нашому випадку, для кешування.

У контексті даної лабораторної роботи було реалізовано кешуючий механізм для системи завантаження файлів, який необхідно було додати, не змінюючи початковий клас SimpleDownloader. Для цього було визначено спільний інтерфейс Downloader (Subject) з єдиним методом download(String url). Клас SimpleDownloader (Real Subject) реалізує цей інтерфейс, імітуючи тривале завантаження файлу.

Для інтеграції кешування було розроблено клас CachingDownloaderProxy (Proxy). Цей клас також реалізує інтерфейс Downloader, що дозволяє йому замінити SimpleDownloader у клієнтському коді. CachingDownloaderProxy містить посилання на об'єкт SimpleDownloader та внутрішню структуру даних (Map) для зберігання кешованих результатів. При виклику методу download(url) Замісник виконує контрольну логіку: спочатку він перевіряє, чи є контент за даним url у кеші. Якщо дані присутні, вони одразу повертаються, і виклик SimpleDownloader не відбувається. Якщо ж дані відсутні, Замісник делегує запит справжньому об'єкту, отримує результат, зберігає його у кеші, і лише потім повертає клієнту.

Такий підхід повністю відповідає принципу відкритості/закритості (Open/Closed Principle), оскільки функціональність було розширено (додано кешування) без зміни початкового класу SimpleDownloader. Клієнтський код взаємодіє лише з інтерфейсом Downloader і не знає, чи працює він із SimpleDownloader безпосередньо, чи з його Замісником, що робить систему гнучкою та легко підтримуваною.

Програмний код реалізації на Java розміщено у Додатку А.

Висновок

Під час виконання лабораторної роботи було успішно вивчено та реалізовано структурний патерн Замісник (Proxy). Завдання інтеграції механізму кешування до класу SimpleDownloader без його модифікації було повністю вирішене.

Клас CachingDownloaderProxy виступив як замісник, реалізувавши той самий інтерфейс Downloader, що дозволило йому прозоро підмінити оригінальний об'єкт для клієнтського коду. Замісник успішно додав додаткову функціональність (кешування), контролюючи доступ до оригінального класу: при першому запиті він викликає SimpleDownloader, а при повторному — повертає дані з внутрішнього кешу, що значно підвищує ефективність та продуктивність системи.

Отримані знання та навички підтверджують ефективність патерна Замісник як засобу для контролю доступу, додавання допоміжної логіки та підвищення продуктивності системи без порушення цілісності та архітектури початкових класів.

Додаток А

Lab6.java