Міністерство освіти і науки України
Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №8
З дисципліни «Технології розроблення програмного забезпечення»
Тема: **«ШАБЛОНИ «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR»»**
Download manager

Виконала:
Студентка групи ІА-22
Степанюк-Боримська А. І.

Перевірив:
Мягкий М. Ю.

Київ-2024

Зміст

Тема:
ШАБЛОНИ «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR»

Мета:
Ознайомитися з основними шаблонами проєктування, такими як «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR», вивчити їхні принципи роботи та навчитись застосовувати для створення гнучкого та масштабованого програмного забезпечення в загальній розробці програмних систем.

Хід роботи

1. Реалізувати не менше 3-х класів відповідно до обраної теми

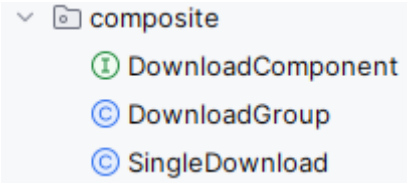

Рис. 1 — Структура проекту

Під час виконання лабораторної роботи було розроблено 3 класи, які реалізують функціонал менеджера завантажень (рис. 1). Нижче наведено детальний опис кожного з цих класів:

1. **DownloadComponent**
   Опис:
   Інтерфейс, який визначає спільні методи для всіх компонентів у структурі Composite. Ці методи включають start(), pause() і showDetails(), які мають виконувати як окремі завантаження, так і групи.
   Призначення:
   Забезпечує єдиний інтерфейс для роботи з окремими завантаженнями (SingleDownload) і групами завантажень (DownloadGroup), дозволяючи обробляти їх однаково.
2. **SingleDownload**
   Опис:
   Клас, що представляє одиничне завантаження. Він реалізує інтерфейс DownloadComponent і надає конкретну реалізацію методів start(), pause(), і showDetails(), які працюють із конкретним об'єктом Download.

Призначення:

Виконує операції над окремим завантаженням, наприклад, запуск, пауза або відображення деталей. Це базовий елемент у структурі Composite.

3. **DownloadGroup**

Опис:

Клас, що представляє групу завантажень. Він реалізує інтерфейс DownloadComponent і зберігає список компонентів (DownloadComponent), які можуть бути як одиничними завантаженнями (SingleDownload), так і іншими групами (DownloadGroup). Реалізує методи start(), pause(), і showDetails() шляхом виклику цих методів для всіх компонентів у групі.

Призначення:

Забезпечує групування завантажень і виконання операцій над усіма елементами в групі одночасно. Дозволяє працювати з вкладеними групами, створюючи ієрархічну структуру завантажень.

2. Реалізувати один з розглянутих шаблонів за обраною темою

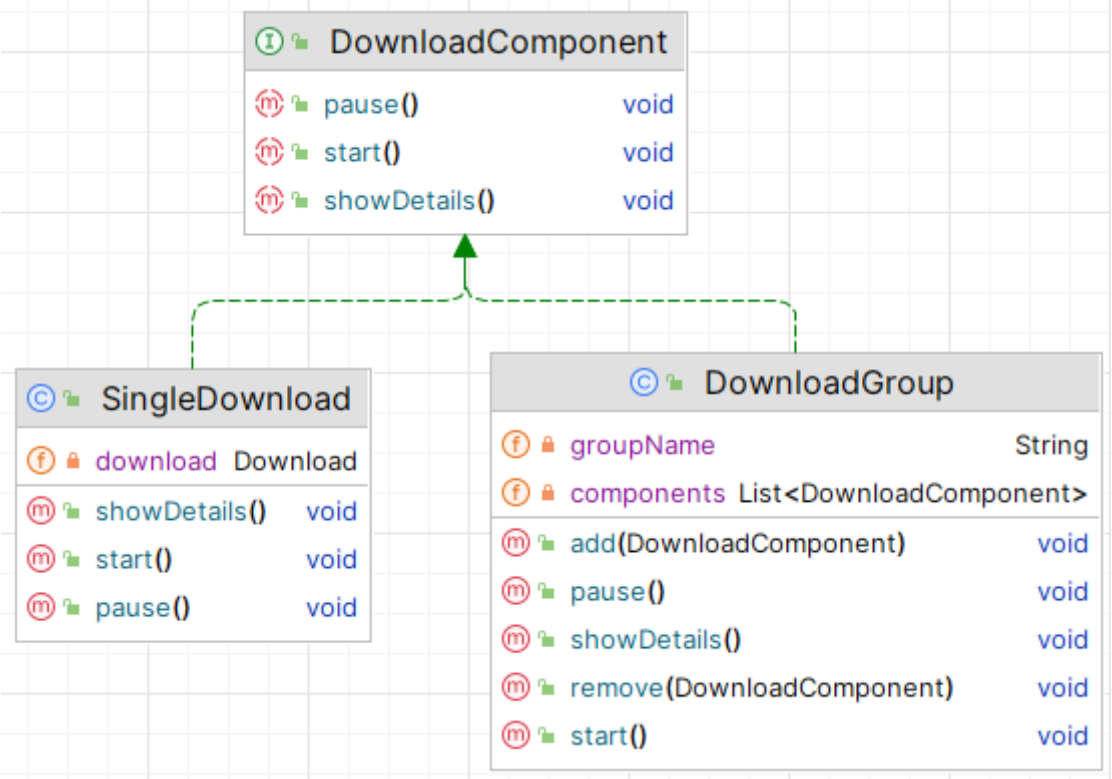

Рис. 2 — Діаграма класів

У проекті менеджера завантажень патерн Composite реалізовано для управління як окремими завантаженнями, так і групами завантажень, використовуючи єдиний інтерфейс. Завдяки цьому можна працювати з одиночними файлами та групами однаково, що робить систему гнучкою та масштабованою.

Як реалізований патерн Composite

1. Інтерфейс DownloadComponent визначає загальні операції (start, pause, showDetails) для всіх компонентів.
2. Клас SingleDownload відповідає за виконання цих операцій для окремого завантаження.
3. Клас DownloadGroup представляє групу завантажень, яка може містити як одиночні завантаження, так і інші групи. Усі операції виконуються рекурсивно для всіх вкладених компонентів, дозволяючи працювати з будь-якою глибиною ієрархії.

Проблеми, які вирішує патерн Composite

1. Управління складними структурами: Завантаження можуть бути організовані у вкладені групи, наприклад, за категоріями чи типами. Composite дозволяє управляти такими структурами просто та ефективно.
2. Уніфікований підхід: Патерн дозволяє однаково працювати з окремими завантаженнями та групами, не ускладнюючи код перевірками типу.
3. Масштабованість: Система легко розширюється — нові типи компонентів (наприклад, специфічні групи з пріоритетами) можна додати, не змінюючи існуючу структуру.
4. Зручність використання: Операції, такі як запуск, пауза або отримання деталей, застосовуються до всіх елементів групи автоматично.

Переваги використання Template Method

1. Модульність: Кожен компонент (SingleDownload або DownloadGroup) відповідає за свою функціональність, що спрощує підтримку коду.
2. Гнучкість: Завдяки рекурсивній природі груп, структура завантажень може бути будь-якої складності.
3. Простота розширення: Нові типи завантажень або груп додаються без змін існуючої логіки.
4. Єдиний інтерфейс: Операції над завантаженнями виконуються однаково, незалежно від того, чи це окремий файл, чи група.

Перевірка роботи

```
public class DownloadManagerApp {
    public static void main(String[] args) {
        Download httpFile = new Download( fileName: "file1.txt",  url: "http://example.com/file1",  fileSize: 1024, DownloadStatus.PENDING,  progress: 0.0);
        Download httpsFile = new Download( fileName: "file2.txt",  url: "https://secure.com/file2",  fileSize: 2048, DownloadStatus.PENDING,  progress: 0.0);
        Download ftpFile = new Download( fileName: "file3.txt",  url: "ftp://ftpserver.com/file3",  fileSize: 4096, DownloadStatus.PENDING,  progress: 0.0);

        SingleDownload httpDownload = new SingleDownload(httpFile);
        SingleDownload httpsDownload = new SingleDownload(httpsFile);
        SingleDownload ftpDownload = new SingleDownload(ftpFile);

        DownloadGroup httpGroup = new DownloadGroup( groupName: "HTTP Downloads");
        httpGroup.add(httpDownload);
        httpGroup.add(httpsDownload);

        DownloadGroup allDownloads = new DownloadGroup( groupName: "All Downloads");
        allDownloads.add(httpGroup);
        allDownloads.add(ftpDownload);

        allDownloads.showDetails();
        allDownloads.start();
        allDownloads.pause();
    }
}
```

Рис. 3 — Перевірка роботи

У цьому прикладі ми створили три окремі завантаження (SingleDownload) і згрупували їх у дві групи (DownloadGroup): одну для HTTP-завантажень і одну для всіх завантажень. Потім викликали методи showDetails, start, і pause для головної групи, щоб перевірити, як ці операції виконуються для всіх елементів, включаючи вкладені групи.

```
Details for group: All Downloads
Details for group: HTTP Downloads
Download details: file1.txt from http://example.com/file1
Download details: file2.txt from https://secure.com/file2
Download details: file3.txt from ftp://ftpserver.com/file3
Starting all downloads in group: All Downloads
Starting all downloads in group: HTTP Downloads
Starting download: file1.txt
Starting download: file2.txt
Starting download: file3.txt
Pausing all downloads in group: All Downloads
Pausing all downloads in group: HTTP Downloads
Pausing download: file1.txt
Pausing download: file2.txt
Pausing download: file3.txt
```

Рис. 4 — Результат роботи

У результаті роботи програми всі завантаження в групах (включаючи вкладені) виконують свої операції. Відображаються деталі файлів, завантаження стартують і переходять у паузу. Це демонструє успішну реалізацію патерну Composite: уніфіковану обробку окремих завантажень і груп.

Висновки:

Реалізація патерну Composite у проекті менеджера завантажень дозволяє ефективно управляти складними структурами завантажень, надаючи гнучкість, масштабованість і простоту у використанні. Це рішення усуває дублювання коду та забезпечує легку інтеграцію нових функцій.