



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 №4
**Технологія розроблення програмного
забезпечення**

*Шаблони «SINGLETON»,
«ITERATOR», «PROXY», «STATE»,
«STRATEGY»*

Варіант 25

Виконав
студент групи ІА-23
Калина С. О.

Перевірив:
Мягкий Михайло
Юрійович

Київ 2024

Мета: Метою даної лабораторної роботи є ознайомлення з шаблонами проєктування, зокрема з шаблоном "Iterator", та їх практичне застосування при розробці програмного забезпечення.

Хід роботи

25. Installer generator (iterator, builder, factory method, bridge, interpreter, client-server)

Генератор інсталяційних пакетів повинен мати якийсь спосіб налаштування файлів, що входять в установку, установки вікон з інтерактивними можливостями (галочка - створити ярлик на робочому столі; ввести в текстове поле деякі дані, наприклад, ліцензійний ключ і т.д.). Генератор повинен вивести один файл .exe або .msi.

Завдання.

- 1 . Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.
3. Застосування одного з розглянутих шаблонів при реалізації програми

Короткі теоретичні відомості.

Шаблони проєктування (або патерни) — це перевірені практикою рішення загальних проблем проєктування програмного забезпечення. Вони описують стандартні підходи, які можна застосувати в різних ситуаціях, щоб вирішити певні проблеми, зберігаючи при цьому структуру системи зрозумілою та підтримуваною. Шаблони проєктування мають загальновживані назви, що дозволяє розробникам легко обговорювати та впроваджувати відповідні підходи.

Основні переваги шаблонів проєктування:

- Зменшення часу та зусиль на створення архітектури.
- Гнучкість та адаптованість системи до змін.
- Полегшення підтримки та розвитку системи.
- Стійкість системи до змін та спрощення інтеграції з іншими системами.

Iterator — це приклад шаблону поведінки, який надає спосіб поелементного проходження по колекції об'єктів без розкриття її внутрішньої структури. Основна ідея шаблону в тому, щоб винести функцію ітерації в окремий клас, дозволяючи колекції зосередитися на зберіганні даних. Це досягається через методи `First()`, `Next()`, `IsDone`, та `CurrentItem`, які дозволяють ініціювати та здійснювати ітерацію по елементах.

Iterator дозволяє створити різні способи обходу колекції без зміни самої колекції. Це дає змогу реалізувати обхід не тільки у порядку, а й, наприклад, у зворотньому порядку, за парними або непарними індексами тощо.

Основні переваги та недоліки шаблону Iterator

Переваги:

- Уніфікація способу обходу для всіх колекцій.
- Можливість реалізації різних способів обходу.
- Спрощення коду зберігання даних.

Недоліки:

- Може бути зайвим для простих колекцій, де підходить звичайний цикл.

Хід роботи

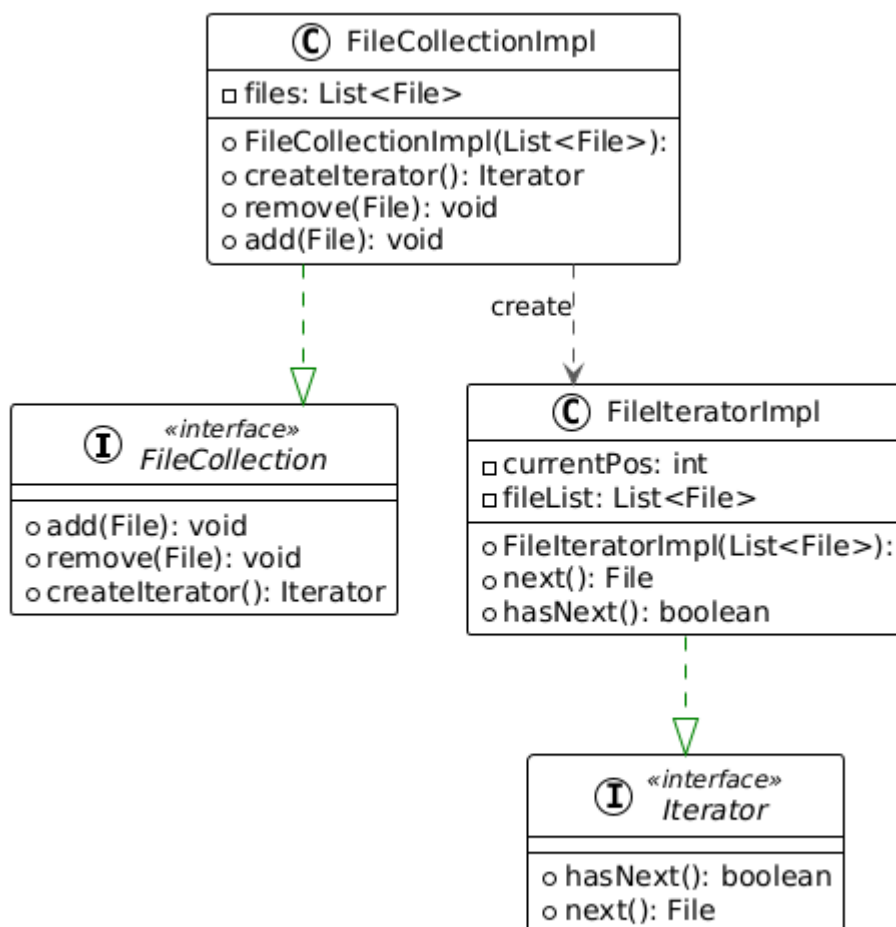


Рисунок №1 – UML діаграма шаблону Ітератор

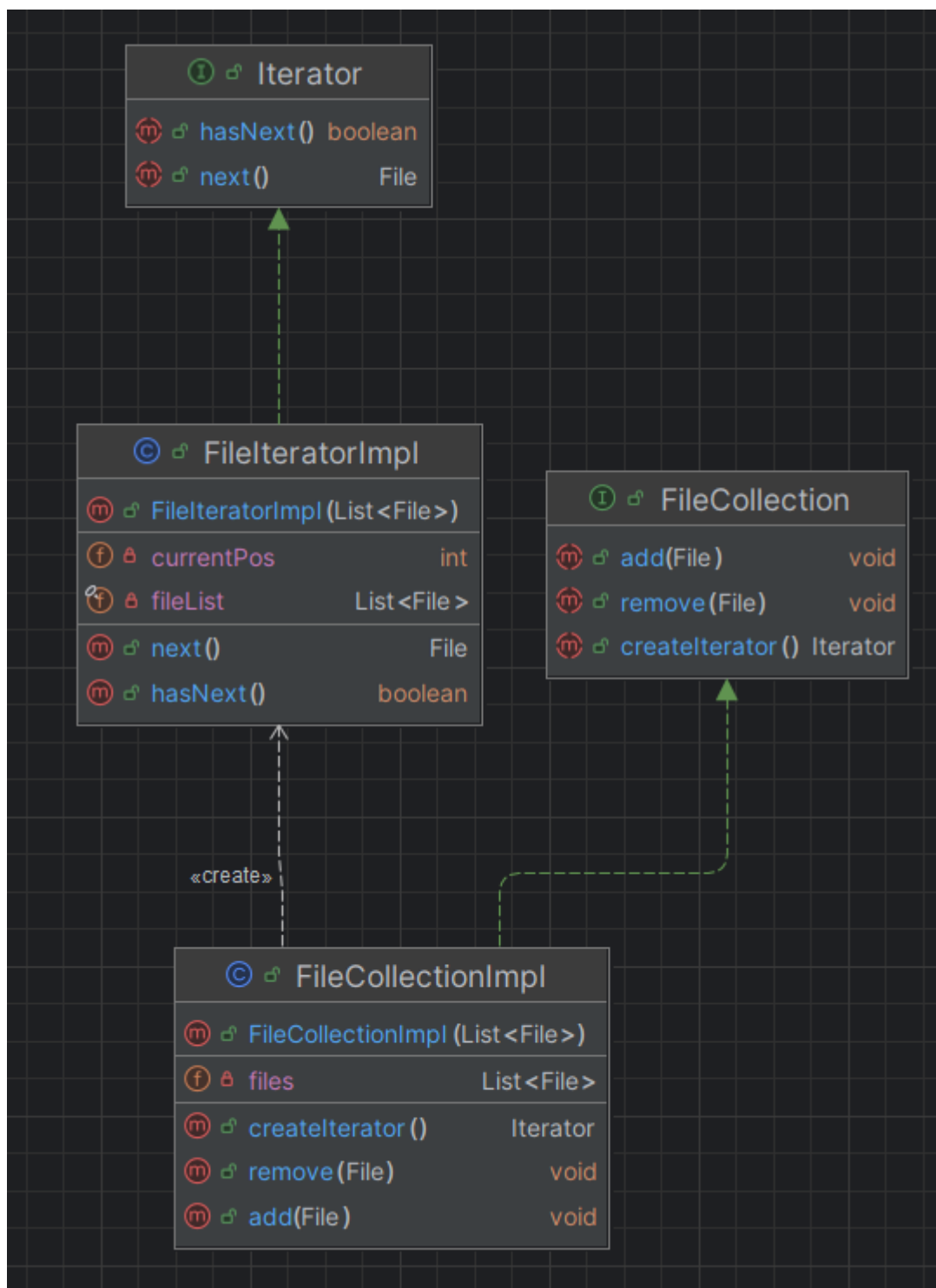


Рисунок №2 – Діаграма класів , згенерована IDE, реалізації шаблону Iterator

На UML-діаграмі показано реалізацію патерну Iterator для роботи з колекцією файлів. У ній є наступні класи:

1. **Iterator** - інтерфейс, який визначає методи hasNext() (перевіряє наявність наступного елемента) і next() (повертає поточний елемент і переходить до наступного).
2. **FileIteratorImpl** - конкретна реалізація інтерфейсу Iterator для роботи з колекцією файлів (List<File>). Має змінні currentPos (позиція поточного елемента) та fileList (список файлів). Реалізує методи hasNext() та next() для послідовного доступу до файлів у колекції.
3. **FileCollection** - інтерфейс, який визначає методи для роботи з колекцією файлів, такі як add() (додає файл до колекції), remove() (видаляє файл) та createIterator() (створює об'єкт ітератора для обходу колекції).
4. **FileCollectionImpl** - конкретна реалізація інтерфейсу FileCollection. Зберігає файли в списку files і реалізує методи add(), remove() та createIterator() для створення екземпляра FileIteratorImpl, який дозволяє обійти файли у колекції.

Робота паттерну

Використаємо прописані об'єкти файлів. Ітератор проходить по списку файлів і виводить інформацію про кожен із файлів.

```

FileName id: 1 size: 1024 fileName: document1.txt path: /user/documents/document1.txt
FileName id: 2 size: 2048 fileName: image1.jpg path: /user/images/image1.jpg
FileName id: 3 size: 4096 fileName: presentation.pptx path: /user/presentations/presentation.pptx
FileName id: 4 size: 8192 fileName: video.mp4 path: /user/videos/video.mp4
FileName id: 5 size: 5120 fileName: music.mp3 path: /user/music/music.mp3

```

Висновок: в результаті виконання лабораторної роботи було реалізовано шаблон проектування "Iterator" для керування колекцією завантажень. Було продемонстровано, як за допомогою ітератора можна проходити по колекції завантажень, отримувати інформацію про кожне завантаження та починати процес завантаження файлу.