

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: «Патерни проектування»

Виконав:
студент групи ІА-32
Задорожний Костянтин
Леонідович

Перевірив:
Мягкий Михайло
Юрійович

Київ - 2025

Тема: Патерни проектування.

Мета: Вивчити структуру шаблонів «Composite», «Flyweight»

(«Пристосуванець»), «Interpreter», «Visitor» та навчитися застосовувати їх в реалізації програмної системи.

Завдання

- Ознайомитись з короткими теоретичними відомостями.
- Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.
- Реалізувати один з розглянутих шаблонів за обраною темою.
- Реалізувати не менше 3-х класів відповідно до обраної теми.
- Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен містити: діаграму класів, яка представляє використання шаблону в реалізації системи, навести фрагменти коду по реалізації цього шаблону.

Предметна область:

22. FTP-server (state, builder, memento, template method, visitor, client-server)

FTP-сервер повинен вміти коректно обробляти і відправляти відповіді п протоколу FTP, з можливістю створення користувачів (з пароллями) 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.

Хід роботи

Для виконання цієї лабораторної роботи я використав патерн Visitor.

Я використав його для генерації звітів по файловій системі користувачів, а саме для підрахунку загального розміру файлів, кількості файлів різних типів без зміни класів самих файлів.

Структура:

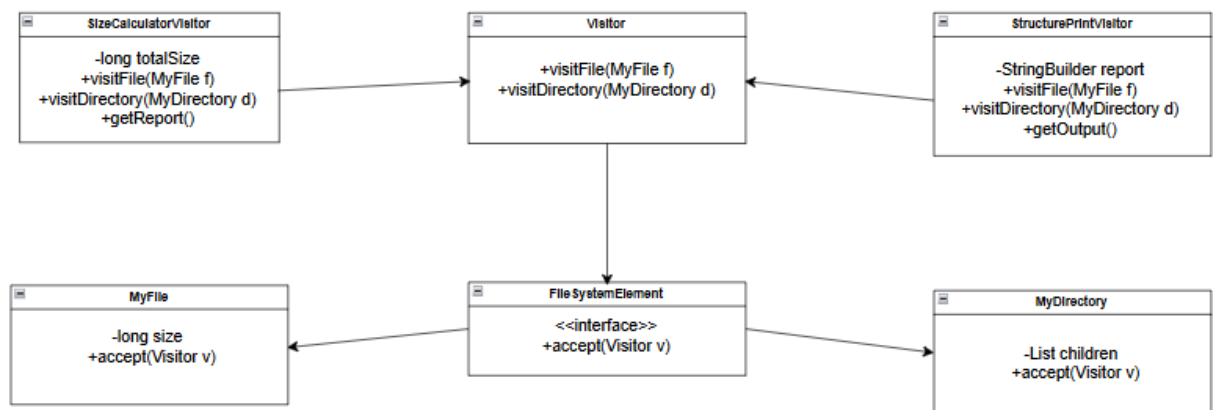
Visitor (Відвідувач): Інтерфейс, що оголошує методи відвідування для кожного конкретного класу елемента (VisitFile, VisitDirectory).

ConcreteVisitor: Реалізує операції підрахунку розміру та генерацію XML-звіту.

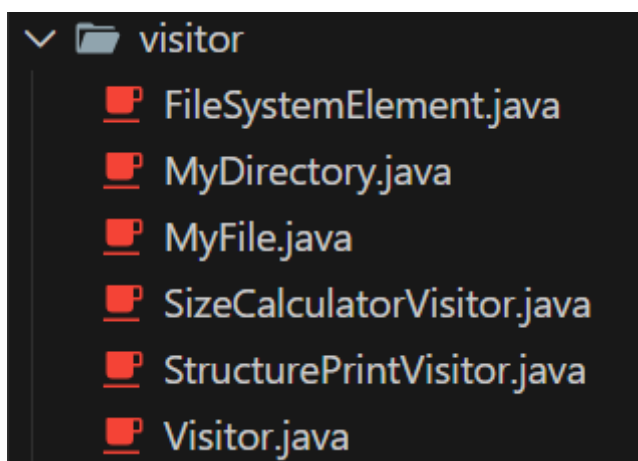
Element: Інтерфейс елемента структури, що оголошує метод асерт.

ConcreteElement: Класи, які реалізують метод асерт, викликаючи відповідний метод відвідувача.

Діаграма структури патерну Visitor для FTP server:



Реалізація в коді:



Інтерфейси (Visitor.java та FileSystemElement.java):

```
src > main > java > com > example > ftp > service > visitor > Visitor.java >
1  package com.example.ftp.service.visitor;
2
3  // Visitor Interface
4  public interface Visitor {
5      void visitFile(MyFile file);
6
7      void visitDirectory(MyDirectory directory);
8  }
```

```
FileSystemElement.java X Visitor.java MyFile.java MyDirectory.java
src > main > java > com > example > ftp > service > visitor > FileSystemElement.java > ...
1  package com.example.ftp.service.visitor;
2
3  public interface FileSystemElement {
4      void accept(Visitor visitor);
5  }
```

Конкретні елементи (MyFile.java, MyDirectory.java)

Ці класи представляють структуру даних. Вони "дозволяють" відвідувачу зайти до них.

```
src > main > java > com > example > ftp > service > visitor > MyDirectory.java > MyDirectory
1  package com.example.ftp.service.visitor;
2
3  import java.io.File;
4  import java.util.ArrayList;
5  import java.util.List;
6
7  public class MyDirectory implements FileSystemElement {
8      private String name;
9      private List<FileSystemElement> children = new ArrayList<>();
10
11     public MyDirectory(File realDirectory) {
12         this.name = realDirectory.getName();
13         // Рекурсивно будуємо дерево
14         File[] files = realDirectory.listFiles();
15         if (files != null) {
16             for (File f : files) {
17                 if (f.isDirectory()) {
18                     children.add(new MyDirectory(f));
19                 } else {
20                     children.add(new MyFile(f));
21                 }
22             }
23         }
24     }
```

```
src > main > java > com > example > ftp > service > visitor > MyFile.java > MyFile
1  package com.example.ftp.service.visitor;
2
3  import java.io.File;
4
5  public class MyFile implements FileSystemElement {
6      private String name;
7      private long size;
8
9      public MyFile(File realFile) {
10         this.name = realFile.getName();
11         this.size = realFile.length();
12     }
13
14     @Override
15     public void accept(Visitor visitor) {
16         visitor.visitFile(this);
17     }
18
19     public String getName() {
20         return name;
21     }
22
23     public long getSize() {
24         return size;
25     }
26 }
```

Конкретні відвідувачі

Реалізація двох різних алгоритмів, які працюють з однією структурою.

1. SizeCalculatorVisitor.java (Рахує загальний розмір файлів):

```
src > main > java > com > example > ftp > service > visitor > SizeCalculatorVisitor.java > ...
1  package com.example.ftp.service.visitor;
2
3  import org.springframework.stereotype.Component;
4
5  @Component
6  public class SizeCalculatorVisitor implements Visitor {
7      private long totalSize = 0;
8      private int fileCount = 0;
9
10     // Скидаємо стан перед новим використанням
11     public void reset() {
12         totalSize = 0;
13         fileCount = 0;
14     }
15
16     @Override
17     public void visitFile(MyFile file) {
18         totalSize += file.getSize();
19         fileCount++;
20     }
21
22     @Override
23     public void visitDirectory(MyDirectory directory) {
24         // Для папки розмір не додаємо, просто враховуємо структуру
25     }
26
27     public String getReport() {
28         return "Total Files: " + fileCount + ", Total Size: " + (totalSize / 1024) + " KB";
29     }
30 }
```

2. StructurePrintVisitor.java (Генерує дерево файлів у вигляді тексту).

```
Visitor.java  MyFile.java  MyDirectory.java  SizeCalculatorVisitor.java  StructurePrintVisitor.java
src > main > java > com > example > ftp > service > visitor > StructurePrintVisitor.java > StructurePrintVisitor > visitDirectory
1  package com.example.ftp.service.visitor;
2
3  import org.springframework.stereotype.Component;
4
5  @Component
6  public class StructurePrintVisitor implements Visitor {
7      private StringBuilder report = new StringBuilder();
8      private int depth = 0;
9
10     public void reset() {
11         report.setLength(newLength: 0);
12         depth = 0;
13     }
14
15     @Override
16     public void visitFile(MyFile file) {
17         indent();
18         report.append(str: "- FILE: ").append(file.getName())
19             .append(str: " (").append(file.getSize()).append(str: " bytes)\n");
20     }
21
22     @Override
23     public void visitDirectory(MyDirectory directory) {
24         indent();
25         report.append(str: "+ DIR: ").append(directory.getName()).append(str: "\n");
26         depth++;
27     }
28
29     private void indent() {
30         for (int i = 0; i < depth; i++) {
31             report.append(str: " ");
32         }
33     }
34 }
```

Результати:

Username	Home Directory	Can Write	Upload Limit	Actions
admin	C:/ftp/admin	true	0	Edit Analyze
Leonid	C:/ftp/user8	true	10000	Edit Analyze
ia-32fiot-23-010	C:/ftp/user8	true	3	Edit Analyze
admin_test	C:/ftp/admin1	true	6	Edit Analyze
adminn	C:/ftp/admin	true	0	Edit Analyze
user1	C:/ftp/user3	true	5000	Edit Analyze

Analysis for user: admin

Disk Usage Report (Visitor 1)

Total Files: 8, Total Size: 31714 KB

File Structure (Visitor 2)

+ DIR: admin

- FILE: 3e6df80eb81b4fe0132e87235b149076de302b... (118949 bytes)

- FILE: lime.ndll (715264 bytes)

- FILE: test.txt (182 bytes)

- FILE: test.txt.txt (78 bytes)

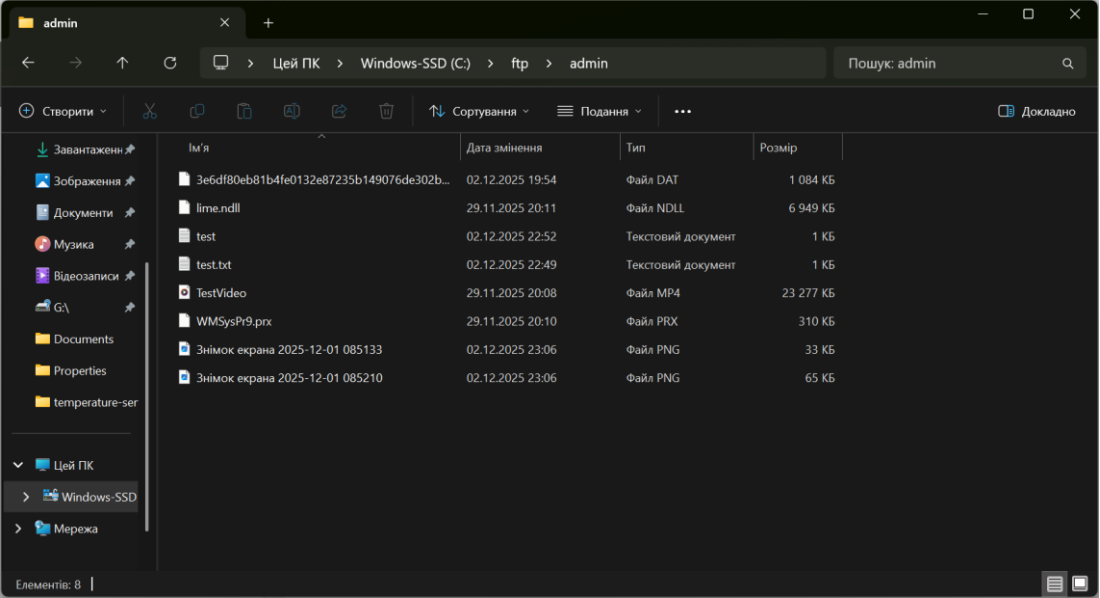
- FILE: TestVideo.mp4 (23854728 bytes)

- FILE: WMSysPr9.prx (316448 bytes)

- FILE: Знімок екрана 2025-12-01 085133.png (32836 bytes)

- FILE: Знімок екрана 2025-12-01 085210.png (66279 bytes)

[Back to Users](#)



Analysis for user: user1

Disk Usage Report (Visitor 1)

Total Files: 2, Total Size: 3416 KB

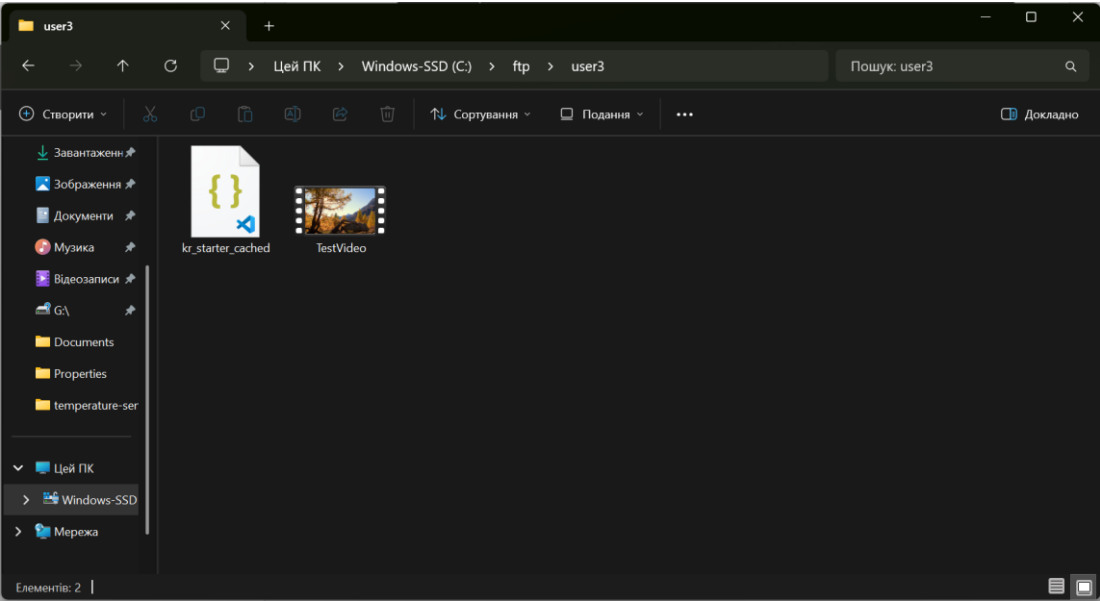
File Structure (Visitor 2)

+ DIR: user3

- FILE: kr_starter_cached.json (46 bytes)

- FILE: TestVideo.mp4 (3407394 bytes)

[Back to Users](#)



На знімках показано результат генерації звітів по файловій системі користувачів для admin та користувача user1. У Disk Usage підрахований SizeCalculatorVisitor, а в Structure - дерево файлів, згенероване StructurePrintVisitor.

Висновки:

Я вивчив структуру шаблонів «Composite», «Flyweight» (Пристосуванець), «Interpreter», «Visitor» та навчився застосовувати їх в реалізації програмної системи.