



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  
**Технологія розроблення програмного  
забезпечення**  
ШАБЛони «COMPOSITE», «FLYWEIGHT»,  
«INTERPRETER», «VISITOR»  
Варіант 8

Виконав  
студент групи ІА-13  
Крутиус Владислав Віталійович

Київ 2023р.

## Шаблон проектування "Composite"

Шаблон проектування "Composite" відноситься до структурних патернів і використовується для об'єднання об'єктів в структури деревоподібної ієрархії для представлення їх як одиничних об'єктів. Цей шаблон дозволяє клієнтам обробляти як одиничні об'єкти, так і композитні структури однаковою способом.

Основні учасники шаблону "Composite":

Component (Компонент): Визначає інтерфейс для всіх конкретних об'єктів та композитів.

Leaf (Листок): Представляє одиничний об'єкт, який не має підкомпонентів.

Composite (Композит): Представляє композит, який має підкомпоненти.

Реалізує методи інтерфейсу Component для роботи з підкомпонентами.

Принцип роботи шаблону полягає в тому, що інтерфейс для індивідуальних об'єктів і композитів є ідентичним, що дозволяє їх використовувати уніфіковано.

## Шаблон проектування "Flyweight"

Шаблон проектування "Flyweight" відноситься до структурних патернів і використовується для оптимізації роботи з об'єктами, які мають спільні частини, шляхом винесення загальної частини в зовнішню структуру.

Цей шаблон дозволяє економити ресурси, особливо коли кількість об'єктів велика і багато з них може використовувати одні й ті ж ресурси.

Основні учасники шаблону "Flyweight":

Flyweight (Легковаговик): Визначає інтерфейс, через який конкретні легковаговики можуть отримати та використовувати зовнішні ресурси.

ConcreteFlyweight (Конкретний легковаговик): Реалізує інтерфейс легковаговика та зберігає внутрішній стан, який може бути спільно використаний.

UnsharedConcreteFlyweight (Неподілений конкретний легковаговик): Має власні внутрішній стан, який не може бути спільно використаний з іншими об'єктами.

FlyweightFactory (Фабрика легковаговиків): Відповідає за управління легковаговиками, забезпечуючи можливість вилучення та повернення легковаговиків.

Принцип роботи шаблону полягає в тому, що спільні ресурси витягуються в окремий об'єкт (легковаговик), і замість створення нового об'єкта для кожного випадку використання ресурсів, використовується вже існуючий об'єкт.

## **Шаблон проектування "Interpreter"**

Шаблон проектування "Interpreter" відноситься до поведінкових патернів і використовується для визначення граматики для мови та інтерпретації речень цієї мови. Цей шаблон дозволяє створити інтерпретатор, який читає, розуміє та виконує вирази, визначені у мові.

Основні учасники шаблону "Interpreter": `AbstractExpression` (Абстрактний вираз): Визначає інтерфейс для інтерпретаторів.

`TerminalExpression` (Термінальний вираз): Реалізує інтерфейс абстрактного виразу для термінальних символів граматики.

`NonterminalExpression` (Нетермінальний вираз): Реалізує інтерфейс абстрактного виразу для нетермінальних символів граматики, об'єднує термінальні вирази та інші нетермінальні вирази.

`Context` (Контекст): Містить інформацію, яку інтерпретатор використовує для виконання операцій.

`Client` (Клієнт): Створює об'єкт контексту та вирази для інтерпретації мови.

Принцип роботи шаблону полягає в тому, що ми визначаємо граматику мови та створюємо вирази для інтерпретації цієї граматики. Клієнт створює об'єкт контексту та вирази, інтерпретатор яких здатен виконувати вирази, представлені у мові.

## **Шаблон проектування "Visitor"**

Шаблон проектування "Visitor" відноситься до поведінкових патернів і використовується для визначення нової операції над об'єктами без зміни їхньої структури. Цей шаблон дозволяє визначати нові алгоритми, не змінюючи класи об'єктів, над якими ці алгоритми використовуються.

Основні учасники шаблону "Visitor":

`Visitor` (Відвідувач): Визначає інтерфейс з методами відвідувача для кожного класу елемента.

`ConcreteVisitor` (Конкретний відвідувач): Реалізує інтерфейс відвідувача та визначає конкретні алгоритми для кожного класу елемента.

Element (Елемент): Визначає інтерфейс для об'єктів, над якими можуть бути використані операції відвідувача.

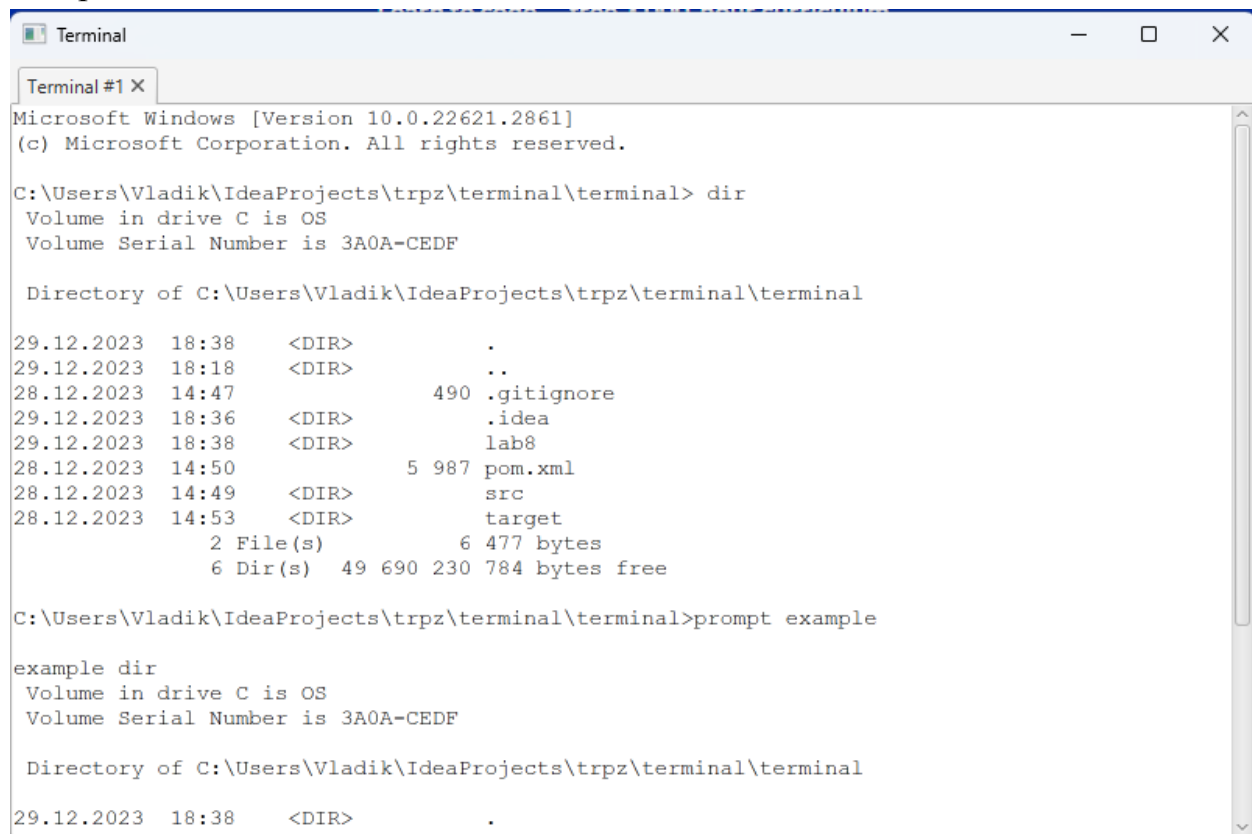
ConcreteElement (Конкретний елемент): Реалізує інтерфейс елемента та визначає специфічну для класу операцію відвідувача.

ObjectStructure (Об'єктна структура): Зберігає колекцію об'єктів і надає інтерфейс для їхньої ітерації.

## Патерн Visitor

Реалізовано в програмі шляхом визначення окремих конкретних команд.

На приклад



```
Terminal
Terminal #1 X
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vladik\IdeaProjects\trpz\terminal\terminal> dir
Volume in drive C is OS
Volume Serial Number is 3A0A-CEDF

Directory of C:\Users\Vladik\IdeaProjects\trpz\terminal\terminal

29.12.2023  18:38    <DIR>          .
29.12.2023  18:18    <DIR>          ..
28.12.2023  14:47             490 .gitignore
29.12.2023  18:36    <DIR>          .idea
29.12.2023  18:38    <DIR>          lab8
28.12.2023  14:50             5 987 pom.xml
28.12.2023  14:49    <DIR>          src
28.12.2023  14:53    <DIR>          target
                2 File(s)            6 477 bytes
                6 Dir(s)  49 690 230 784 bytes free

C:\Users\Vladik\IdeaProjects\trpz\terminal\terminal>prompt example

example dir
Volume in drive C is OS
Volume Serial Number is 3A0A-CEDF

Directory of C:\Users\Vladik\IdeaProjects\trpz\terminal\terminal

29.12.2023  18:38    <DIR>          .
```