****

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  
**ШАБЛОНИ «SINGLETON»,**

**«ITERATOR», «PROXY», «STATE»,**

**«STRATEGY»**

Варіант 14

Виконала Перевірив:

студентка групи ІА – 13: Мягкий М. Ю.

Михайленко Андрій

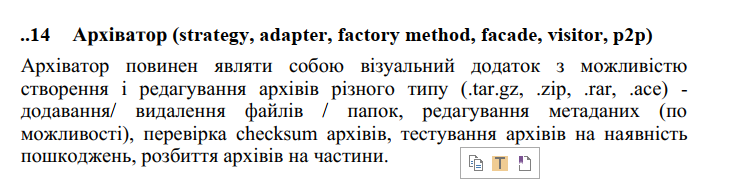
**Завдання:**

1. Ознайомитися з короткими теоретичними відомостями.

2. Реалізувати частину функціонала робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.

3. Застосування одного з розглянутих шаблонів при реалізації програми

**Варіант:**



**Хід роботи**

**Паттерн Стратегія**

Патерн використовується для обробки операцій над різними типами архівних файлів (наприклад, .tar.gz, .zip, .rar і .ace).

Розглянемо, як застосовується паттерн "Стратегія":

Інтерфейс стратегії (ArchiveStrategy): Це абстрактний базовий клас, який визначає спільний інтерфейс для всіх конкретних стратегій. Він оголошує такі методи, як create, extract, add, remove, edit\_metadata, show\_metadata і test. Кожен метод приймає archive\_manager та інші параметри, що стосуються операції.

Конкретні стратегії (TarGzStrategy, ZipStrategy, RarStrategy, AceStrategy): Ці класи реалізують інтерфейс ArchiveStrategy, забезпечуючи специфічну поведінку для роботи з різними форматами архівів. Наприклад, TarGzStrategy містить логіку створення, вилучення, додавання файлів та інших операцій, характерних для архівів .tar.gz.

Контекст (ArchiveManager): Цей клас використовується для взаємодії зі стратегіями. Він містить посилання на об'єкт стратегії та делегує йому виконання операцій. Стратегія, яку слід використовувати, визначається на основі типу архіву (наприклад, .tar.gz, .zip і т.д.).

Адаптери (TarGzAdapter, ZipAdapter, RarAdapter, AceAdapter): Ці класи діють як адаптери між ArchiveStrategy та реальними операціями. Вони беруть об'єкт стратегії у своєму конструкторі і делегують виклики операцій цій стратегії. Це дозволяє використовувати стратегії більш гнучко і взаємозамінно.

Висновок: ознайомився з короткими теоретичними відомостями. Реалізував частину функціонала робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей. Застосував паттерн Стратегія для реалізації програми.