****

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 №5  
**ШАБЛОНИ «ADAPTER»,**

**«COMMAND, «CHAIN OF RESPONSIBILITY», «PROTOTYPE»**

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:**

14. Архіватор (strategy, adapter, factory method, facade, visitor, p2p)Архіватор повинен являти собою візуальний додаток з можливістю створення і редагування архівів різного типу (.tar.gz, .zip, .rar, .ace) - додавання/ видалення файлів / папок, редагування метаданих (по можливості), перевірка checksum архівів, тестування архівів на наявність пошкоджень, розбиття архівів на частини.

**Хід роботи**

Паттерн Адаптер використовується для інтеграції різних стратегій обробки архівів (таких як TAR.GZ, ZIP, RAR, ACE) зі спільним інтерфейсом, що дозволяє основній програмі взаємодіяти з різними типами архівів за допомогою єдиного інтерфейсу. Цей патерн особливо корисний у сценаріях, коли програма повинна підтримувати кілька типів взаємозамінної поведінки або алгоритмів.

Розглянемо, як паттерн Адаптер реалізовано:

Структура шаблону адаптера

Цільовий інтерфейс (ArchiveStrategy): Це загальний інтерфейс, який визначає операції для управління архівами. Це абстрактний клас з такими методами, як create, extract, add, remove, edit\_metadata, show\_metadata і test.

Адаптери (TarGzAdapter, ZipAdapter, RarAdapter, AceAdapter): Це конкретні реалізації інтерфейсу ArchiveStrategy. Кожен адаптер обгортає певну стратегію (наприклад, TarGzStrategy, ZipStrategy тощо) і транслює виклики методів інтерфейсу у конкретні операції обгорнутої стратегії.

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