

## ЕФЕКТИВНЕ ВИКОРИСТАННЯ АРХІТЕКТУРНИХ ШАБЛОНІВ ПРОГРАМУВАННЯ ДЛЯ ДОТРИМАННЯ ПРИНЦИПІВ SOLID

Ярошевич Р.О., Бахмацький Б.С.

Харківський національний університет радіоелектроніки, Харків, Україна

Побудова великих систем та комп'ютерних програм вимагає дотримання певних правил для ефективної роботи та взаємодії різних частин команди. Як відомо, загальноприйнятою системою написання коду є SOLID, яка об'єднує в собі наступні принципи: принцип єдиного обов'язку класів, принцип відкритості та закритості при модифікації програмного продукту, принцип заміщення Лісков, принцип розділення інтерфейсів і принцип інверсії залежностей. [1]. Дані правила утворюють єдину систему, яка дозволяє великим командам будувати ефективні додатки, які привносять читабельність та надають можливість безпечної модифікації коду, що не понесе за собою великих збоїв застосунку. Дані принципи дозволяють досягти цього завдяки сегрегації великих ієрархій на менші частини та об'єднання їх під один керуючий процесом інтерфейс.

Окрім цього, дані правила вводять певні обмеження для програміста, оскільки вони чітко прописують, які прийоми не рекомендується використовувати при написанні програмного коду задля уникнення ситуацій, коли необхідно перебудувати всю ієрархію у разі незначних змін. Проте виникає складність, яка полягає в нерозумінні багатьох початківців ефективно застосувати їх на практиці.

**Метою доповіді** є визначити, яким чином можна імплементувати SOLID – принципи у програмний продукт за допомогою використання практичних і загально відомих архітектурних шаблонів програмування [2].

В доповіді наводяться приклади реалізації згаданих вище SOLID-принципів на основі усіх типів архітектурних шаблонів: породжувальних, структурних і поведінкових. З метою максимально ефективно продемонструвати роботу шаблонів, за основу взято реальний проект, який являє собою застосунок-вікторину як із тривіальними UI-елементами у вигляді таблиць та списків, так і з динамічним ігровим процесом, що поєднує в собі різноманітний функціонал.

Також продемонстровано різницю однотипних архітектурних шаблонів та їх відмінності (різниця між «Фабрикою» та «Будівельником», або між «Станом» та «Стратегією»).

### Список літератури

1. «Чому SOLID – важлива складова мислення програміста» – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/solid-principles/> – 21.03.2024 р. Загол. з екрану.
2. Еріх Гамма, Річард Хелм, Ральф Джонсон, Джон Вліссідес. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994. 395 с.