

УДК 004.853

**Дубчак А.О., д.т.н., проф. Литвиненко Я.В.**

(Тернопільський національний технічний університет імені Івана Пулюя)

## **ПАТТЕРНИ ТА ПРИНЦИПИ. ПАНАЦЕЯ ЧИ ПРОБЛЕМА**

UDC 004.853

**Dubchak A.O., Dr., Prof. Lytvynenko I.V.**

## **PATTERNS AND PRINCIPLES. PANACEA OR PROBLEM**

Розробка програмних продуктів передбачає проектування систем, які дозволяють розширюватись та замінити її елементи з мінімальними затратами часу та ресурсів. Конкуренція на ринку програмних продуктів зумовлює потребу у використанні певних шаблонів, або паттернів. Проте, ці шаблонні рішення можуть створювати більше проблем, ніж вирішувати.

Дана доповідь присвячена проблемі використання паттернів та принципів у розробці програм.

Одне із перших відомих видань, які сформувавши принципи розробки програмних продуктів були описані в так званих GoF принципах[1]. Дані принципи описують загальні структури, які використовуються на різних архітектурних рівнях, що дозволяють розширювати та змінювати програму без значних зусиль. Багато загальноприйнятих підходів були об'єднані у даних принципах, наприклад принцип «проксі». Даний шаблон проектування говорить про сутність, яка може з'єднувати дві інші сутності, коли вони не мають прямого сполучення.

Іншим важливим набором шаблонів є так званий «SOLID»[2]. Це набір із п'яти принципів, які описують і деталізують принципи побудови високорівневої архітектури програм, а також легко проектується на сутності об'єктно-орієнтованого програмування у вигляді класів. Ці принципи є доволі простими та говорять про особливості та властивості елементів системи для її правильного проектування. Наприклад, перший принцип говорить про «Single Responsibility», або про «Єдине призначення». Він говорить про те, що кожна сутність в програмному продукті має виконувати лише одну задачу. Наприклад, якщо ми проектуємо умовну структуру університету, і виділяємо різні ролі, то варто наділяти конкретними задачами різні сутності. Коли нам потрібно описати ролі бухгалтера, викладача та студента, не варто створювати одну сутність по типу «персона», та наділяти її методами калькуляції заробітної плати, викладання лекцій та перевірка лабораторних робіт, та відвідування лекцій та виконання лабораторних робіт. Ці сутності варто розділяти на 3 конкретні ролі, які мають лише необхідний функціонал.

Зчасту, дані принципи ігноруються розробниками-початківцями, а також принципи та паттерни дуже рідко включаються у навчальні програми та курси.

Підсумовуючи, існує багато загальноприйнятих паттернів та принципів, та критичним є питання їх популяризація в середовищі розробників. В даній доповіді детально розглянуті основні ідеї шаблонної розробки та підсумовано питання їх вивчення та використання.

Література.

1. Design Patterns: Elements of Reusable Object-Oriented / E.Gamma, R. Helm, R. Johnson, J. Vlissides., 1994. – 416 с. – (1st edition). – (ISBN-13 : 978-0201633610).
2. Martin R. Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) / Robert Martin., 2017. – 432 с. – (1st). – (ISBN-13 : 978-0134494166).