УДК 004.4

С. Р. Бузиновська

АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО ПРОЄКТУВАННЯ МОДУЛЬНИХ ПРОГРАМНИХ ДОДАТКІВ

Вінницький національний технічний університет

Анотація

У роботі проаналізовано сучасні підходи до проєктування модульних програмних додатків. Розглянуто основні принципи модульної архітектури, зокрема поділ функціональності на незалежні компоненти, слабку зв'язаність та повторне використання коду. Досліджено актуальні архітектурні підходи, такі як компонентно-орієнтована архітектура, Composable Architecture та Modular Monolith. Показано, що застосування модульного підходу сприяє підвищенню масштабованості, зручності супроводження та якості програмного забезпечення.

Ключові слова: програмні додатки, Composable Architecture, Modular Monolith, повторне використання коду.

Abstract

The paper analyzes modern approaches to designing modular software applications. The main principles of modular architecture are considered the division of functionality into independent components, loose coupling, and code reuse. Current architectural approaches are studied, such as component-oriented architecture, Composable Architecture, and Modular Monolith. It is shown that the use of a modular approach contributes to increasing scalability, maintainability, and software quality.

Keywords: software applications, Composable Architecture, Modular Monolith, code reuse.

Вступ

У сучасних умовах розвитку інформаційних технологій програмні системи стають дедалі складнішими, масштабнішими та більш вимогливими до підтримки й розширення. Традиційні монолітні підходи до розробки програмного забезпечення часто ускладнюють процеси модифікації, тестування та повторного використання коду. У зв'язку з цим все більшої актуальності набуває використання модульного підходу до проєктування програмних додатків [1]. Метою даної роботи є аналіз сучасних підходів до проєктування модульних програмних додатків, а також визначення їх переваг і особливостей застосування в процесі розробки програмного забезпечення.

Результати дослідження

Модульна програма — це програмний додаток, архітектура якого базується на поділі функціональності на незалежні модулі. Кожен модуль відповідає за реалізацію конкретного набору функцій і може розроблятися, тестуватися та модифікуватися незалежно від інших компонентів системи.

Основними принципами модульної архітектури є слабка зв'язаність між модулями, однак висока всередині модуля, чітко визначені інтерфейси взаємодії, можливість повторного використання програмних компонентів. Дотримання цих принципів сприяє підвищенню якості програмного забезпечення, зменшенню кількості помилок та полегшенню командної розробки.

Загальну концепцію модульної архітектури представлено на рисунку 1. На схемі зображено дворівневу структуру програмної системи, де верхній рівень представлений функціональними компонентами, а нижній — програмними модулями, що реалізують відповідну функціональність через визначені інтерфейси [1].

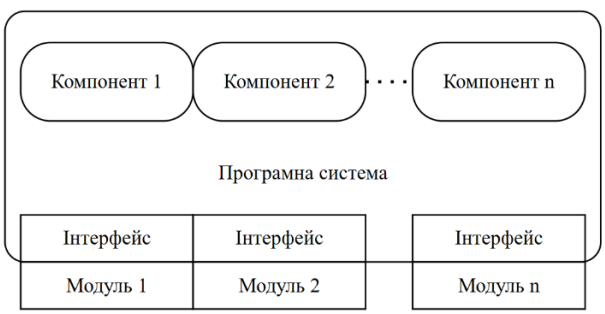

Рисунок 1 – Модульна архітектура

На сьогодні існує декілька поширених підходів до реалізації модульності у програмних системах. Одним із класичних варіантів є компонентно-орієнтований підхід, за якого програма складається з автономних компонентів із чітко визначеними ролями.

Компонентно-орієнтована архітектура передбачає побудову програмного додатка у вигляді набору незалежних компонентів, кожен з яких інкапсулює певну частину функціональності та має чітко визначений інтерфейс взаємодії. Такий підхід базується на принципах повторного використання програмних компонентів, ізоляції змін та розділення відповідальностей, що сприяє підвищенню якості програмного забезпечення та зменшенню складності системи загалом.

Окрему увагу в сучасних дослідженнях приділено концепції Composable Architecture, яка є сучасним підходом до проєктування програмних систем, що орієнтований на гнучке комбінування незалежних модулів і сервісів у єдину систему [2]. Основна ідея полягає в тому, що програмний додаток складається з автономних функціональних блоків, які можуть динамічно поєднуватися між собою відповідно до поточних вимог. Такий підхід забезпечує високу адаптивність програмних систем до змін бізнес-логіки та технологічного середовища.

Важливою особливістю Composable Architecture є використання чітко визначених інтерфейсів, API-first підходу та стандартизованих механізмів взаємодії між модулями. Це дозволяє інтегрувати різні технології та платформи в межах одного програмного рішення, зменшуючи залежність між компонентами. У результаті підвищується масштабованість системи, спрощується її модернізація та скорочується час впровадження нових функціональних можливостей.

Також актуальним є підхід Modular Monolith, що поєднує переваги традиційної монолітної архітектури та принципи модульності [3]. У межах цього підходу програмний додаток реалізується як єдина система, проте його внутрішня структура поділена на логічно незалежні модулі з чітко визначеними межами відповідальності. Кожен модуль інкапсулює власну бізнес-логіку та взаємодіє з іншими модулями лише через публічні інтерфейси.

Застосування Modular Monolith дозволяє уникнути складнощів, характерних для розподілених систем, зокрема проблем синхронізації, мережевих затримок і складного розгортання. Водночас зберігаються ключові переваги модульної архітектури — зручність супроводження, тестування та можливість подальшого переходу до мікросервісної архітектури. Саме тому ця концепція розглядається як ефективний компроміс між простотою реалізації та масштабованістю сучасних програмних систем.

На практиці модульний підхід реалізується за допомогою різноманітних архітектурних патернів, зокрема MVC, MVVM, Clean Architecture та інших. Значну роль відіграють також сучасні засоби автоматизації збірки, тестування та безперервної інтеграції, які сприяють підтримці незалежності модулів.

Для аналізу якості модульної структури програмних додатків застосовуються спеціальні метрики та інструменти, що дозволяють оцінити рівень зв’язаності, складність взаємодії між модулями та потенційні архітектурні проблеми.

Висновки

Проведений аналіз показав, що модульний підхід до проєктування програмних додатків є одним із ключових напрямів розвитку сучасного програмного забезпечення. Використання модульної архітектури дозволяє підвищити масштабованість, зручність супроводження та якість програмних систем.

Сучасні підходи, такі як Composable Architecture та Modular Monolith, поєднують теоретичні принципи модульності з практичними потребами розробки, що робить їх перспективними для подальших досліджень і застосування у навчальних та промислових проєктах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Mbugua S. T., Korongo J., Mbuguah S. On Software Modular Architecture: Concepts, Metrics and Trends // International Journal of Computer & Organization Trends. – Vol. 12, No. 1, 2022. – P. 3-10. – Режим доступу: https://ijcotjournal.org/volume-12-issue-1/IJCOT-V12I1P302.pdf
2. Rusum G. P., Anasuri S. Composable Enterprise Architecture: A New Paradigm for Modular Software Design // International Journal of Emerging Research in Engineering and Technology. – Vol. 4, No. 1, 2023. – P. 99-111. – Режим доступу: https://ijeret.org/index.php/ijeret/article/view/271/258
3. Su R., Li X. Modular Monolith: Is This the Trend in Software Architecture? – arXiv, 2024. – Режим доступу: https://arxiv.org/pdf/2401.11867

Бузиновська Софія Русланівна – студентка групи 1КІ-23б, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, email: sophiabuzynovska@gmail.com.

Buzynovska Sofiia R. – student, 1CE-23b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: sophiabuzynovska@gmail.com.