

УДК 004.021

ДОСЛІДЖЕННЯ ONION-АРХІТЕКТУРИ У ПРЕДМЕТНО-ОРІЄНТОВАНОМУ ПІДХОДІ

В.О. Мамонтов, аспірант

Київський національний університет технологій та дизайну

О.Я. Ніконов, доктор технічних наук, професор

Київський національний університет технологій та дизайну

Ключові слова: архітектура, ядро додатку, модель предметної області, служби домену, служби додатків, інфраструктура.

Існує багато аспектів архітектури Onion, але головна мета полягає в тому, як вона контролює зв'язокі взаємодію з іншими компонентами системи. Головне правило, зводиться до того, що всі з'єднання спрямовані до центру. На рис. 1 представлено приклад Onion архітектури. Аналізуючи Onion архітектуру можна зробити наступні твердження:

- Рівень «Інфраструктура» має доступ до «Служби додатків», «Служби домену» та «Модель домену».
- «Служби додатків» має доступ до «Служби домену» та «Модель домену».
- «Доменні служби» має доступ до «Доменну модель».
- «Доменна модель» має доступ лише до себе.

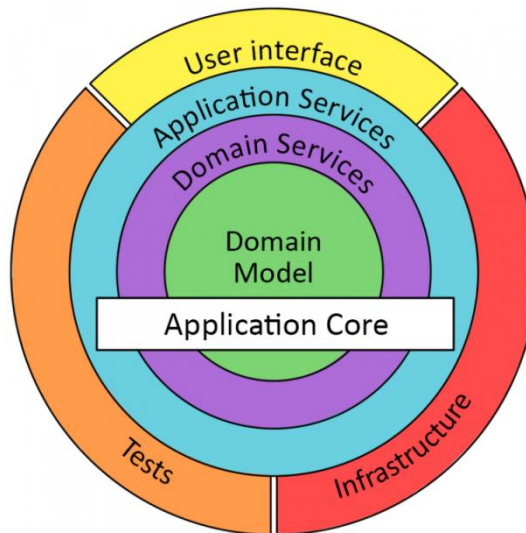


Рисунок 1 – Onion архітектура

Один із підходів до успішної реалізації архітектури програмного забезпечення, зокрема моделі предметної області, складається з забезпечення їй повної ізоляції від інших компонентів системи, що дозволяє досягнути кількох суттєвих переваг: спрощення розуміння та тестування кодової бази, зниження когнітивного навантаження для розробників і полегшення розробки великих та складних проєктів [1].

Ізольована модель предметної області допомагає визначити межі її функціональності та взаємодії з іншими компонентами системи, що робить розуміння та управління складністю проєкту більш ефективним.

Модель сприяє розробці коду, який більш чистий та доречний для відображення бізнес-логіки, що дозволяє підтримувати і покращувати код легше та без ризику негативного впливу на інші частини системи.

Ізольована модель предметної області робить процес модульного тестування більш очевидним та нескладним, завдяки незалежному тестуванню від інших компонентів системи.

Ізоляція предметної області дозволяє вносити зміни та розширювати функціональність без впливу на інші частини системи, що сприяє гнучкості та масштабованості проєкту.

На рис. 2 наведено діаграму сутностей. Один з нескладних способів визначити повну ізоляцію моделі предметної області полягає у створенні діаграми сутностей та об'єктів значень разом із класами з навколишнього середовища. Якщо ці сутності та об'єкти значень не взаємодіють з будь-якими класами, які не є частиною моделі предметної області, то можна стверджувати, що модель предметної області успішно ізольована від зовнішнього середовища [2]. Це спрощує підтримку моделі та допомагає знизити вплив зовнішніх факторів на її функціонування.

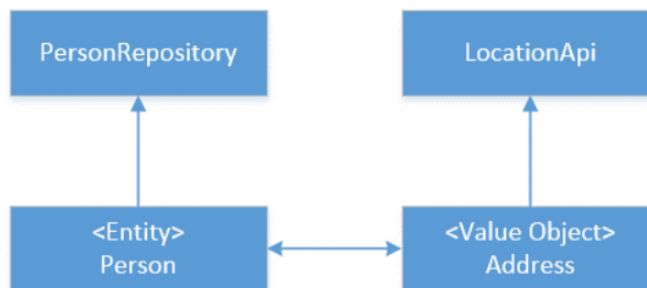


Рисунок 2 – Діаграма сутностей

Результати проведеного дослідження Onion-архітектури у предметно-орієнтованому підході показали, що ізоляція моделі предметної області є ключовим елементом в досягненні успішної реалізації складних програмних проєктів та забезпеченні їх розуміння, підтримки та розвитку.

Список використаних джерел

1. Domain-Driven Design на практиці [Електронний ресурс]: матеріали вебінара. Режим доступу: <https://careers.epam.ua/events/java-webinar-03022022> (дата звернення 20.10.22). – Назва з екрана.

2. Laribee D. Best Practice – An Introduction To Domain - Driven Design [Electronic resource]: article/ D. Laribee. – Accessmode:<https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/best-practice-an-introduction-to-domain-driven-design> (last access: 22.01.2016). – Title from the screen.