

# ПРИНЦИПИ ПОБУДОВИ МУЛЬТИПЛАТФОРМНОЇ БІЗНЕС-ЛОГІКИ ДЛЯ МОБІЛЬНИХ ДОДАТКІВ ЗА ДОПОМОГОЮ KOTLIN MULTIPLATFORM

Вінницький національний технічний університет

## Анотація

*Розглянуто принципи побудови мультиплатформеної бізнес-логіки для пришвидшення розробки нативних додатків під мобільні платформи (Android, iOS) з використанням технологій: Kotlin Multiplatform, Kodein, Ktor, Kotlin Coroutines.*

**Ключові слова:** *мультиплатформа, Kotlin Multiplatform, Android, iOS, DI, мобільні платформи, Kotlin.*

## Abstract

*Paper contains description of principles of accelerated development of native applications for mobile platforms (Android, iOS) with multi-platform business logic using technologies: Kotlin Multiplatform, Kodein, Ktor, Kotlin Coroutines.*

**Keywords:** *multiplatform, Kotlin Multiplatform, Android, iOS, DI, mobile platforms, Kotlin.*

## Вступ

Близько 5.22 мільярдів людей мають мобільні пристрої, що становить близько 67% населення планети[1]. Це дає можливість розробникам програмного забезпечення для мобільних пристроїв адресувати більш широкую аудиторію, отримувати більше коштів. В той же час це призвело до значного зростання конкуренції і підвищило рівень вимог до мобільних застосунків, як, наприклад, до зручності їх використання, якості їх дизайну, а також пришвидшило розвиток мов програмування для написання мобільних програмних продуктів (Kotlin, Swift, Dart, C# та інші) та технологій розробки під мобільні платформи. Загалом, стрімкий розвиток сфери має як і позитивні, так і негативні наслідки. Адже приріст аудиторії відкриває великі можливості для бізнесу, і тим самим завищує потенціал, очікуваний від нових мобільних застосунків, які потрібно створити за мінімальний відрізок часу. З другого боку, через відмінності різних мобільних платформ виникає проблема врахування цих відмінностей при функціонуванні бізнес-логіки, щоб користувач отримував однаково якісний сервіс незалежно від використаної мобільної платформи.

Метою роботи є дослідження принципів побудови бізнес-логіки проекту за допомогою Kotlin Multiplatform для мобільних платформ Android та iOS із використанням сучасних підходів до побудови архітектури та створення програмного продукту.

Об'єктом дослідження є процес створення мультиплатформеної бізнес-логіки для мобільних додатків за допомогою Kotlin Multiplatform.

Предметом дослідження є засоби програмування та побудови мультиплатформеної бізнес-логіки з використанням наступних мов програмування та технологій: Kotlin, Swift, Kotlin Multiplatform, Gradle, XML, JSON, Kodein, Ktor, Kotlin Coroutines.

Частина проекту, яка містить в собі реалізацію бізнес-логіки, а саме взаємодію з сервером та базою даних, логіку обрахунків (якщо вони робляються безпосередньо в додатку) чи логіку структуризації даних та взаємодії з ними, написана за допомогою Kotlin Multiplatform, розділена на 3 модулі[2]. Схему взаємодії модулів подано на рисунку 1.

Основним із трьох модулів є Share [3], у якому прописується спільна логіка для двох мобільних платформ. У ньому найчастіше реалізують взаємодію з сервером та перетворення необроблених масивів даних, які надаються безпосередньо веб-сервісом, у необхідний формат для відображення на графічному користувацькому інтерфейсі. Наприклад, у наведеному модулі доречною буде реалізація пошуку та сортування, конвертація часових даних у потрібний формат та інше.

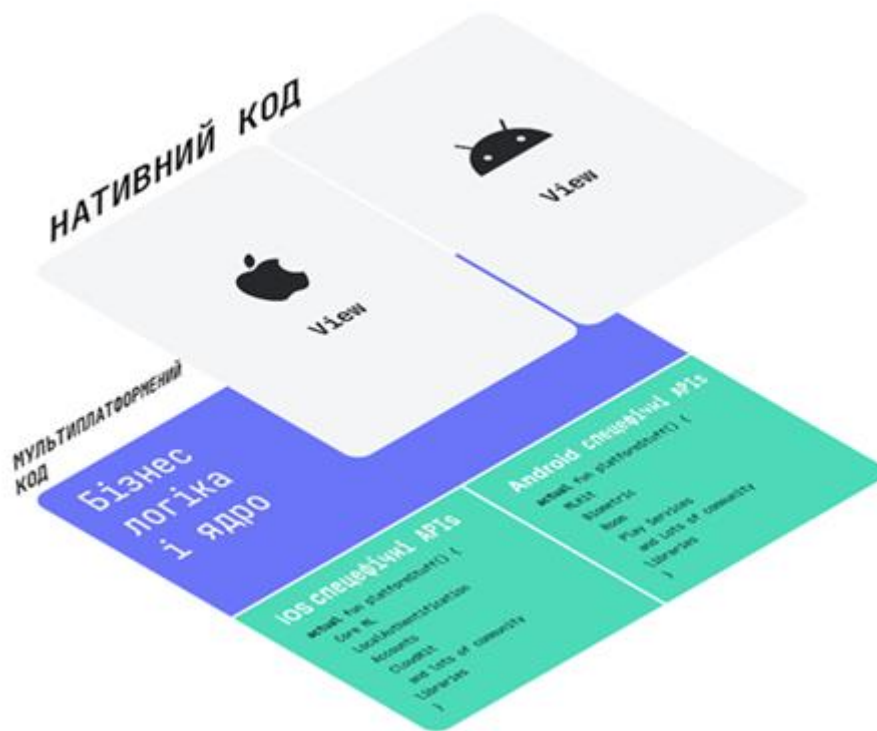


Рисунок 1 — Схема взаємодії ядра та бізнес-логіки, написаної за допомогою Kotlin Multiplatform та нативних додатків

Також у Shared модулі прописується ядро нашого DI (dependency injection, ін'єкції залежностей). Розберемо принцип написання частини, яка взаємодіє з серверною стороною, та структуру правильно створеного DI.

1. Частину для взаємодії з сервером та перетворення даних у форматі JSON у зручні для використання на мові програмування Kotlin, data class-и або списки цих data class-ів. Для простого та швидкого написання ми будемо використовувати бібліотеку Ktor [4], у якій вже прописана вся логіка роботи з сервером побудованим за принципами REST.
2. DI нашої бізнес логіки слід будувати за допомогою сучасної бібліотеки Kodein [5], що допомагає побудувати швидко в роботі та просту у розширенні ін'єкція залежностей. Для правильного створення DI потрібно розділити всі компоненти на модулі, так щоб вони мали якомога менше відповідальностей. А також забезпечити відсутність явних залежностей модулів один від одного.

До наступних двох модулів належать модуль AndroidApp та iOSApp. У них прописується бізнес логіка, яка має бути відповідною до використовуваної платформи, а також складові, що мають різні підходи чи технології реалізації. Візьмемо, як приклад, роботу з штучним інтелектом, яка на платформах Android та iOS, не тільки вимагає різної реалізацію підходу, але й використовує зовсім інші технології. Тобто для роботи штучного інтелекту у додатках під операційну систему iOS використовується Core ML, що є розробкою компанії Apple, а для відтворення того самого штучного інтелекту для ОС Android використовується ML Kit, розроблений компанією Google. Отже, у модулях AndroidApp та iOSApp має міститись бізнес-логіка, яка з вищенаведених причин не може бути прописана в спільному модулі Shared.

На сьогодні існує кілька популярних підходів для написання мультиплатформених рішень, а саме бібліотека Flutter і розроблена під неї мова програмування Dart [6], та фреймворк React Native [7] для мови програмування JavaScript. На перший погляд може здатися що ці аналоги є набагато зручнішими так як ти пишеш не тільки бізнес-логіку для двох платформ, а також й інтерфейс. Проте ці рішення не є нативними, що не тільки сповільнює роботу мобільних додатків, а також сильно обмежує роботу з компонентами телефона, такими як Bluetooth, NFC, та іншими. Також при використанні цих підходів розробник повністю не використовує особливості мобільних платформ, їх встроєні бібліотеки, системні особливості, а також ігнорує вимоги дизайнера для обох платформ..

## Висновок

В роботі було розглянуто застосування Kotlin Multiplatform для виконання стандартної бізнес-логіки, що є базовою для більшості сучасних додатків. Проте дана технологія дозволяє реалізовувати набагато складніші системи, із тривіальними математичними операціями, взаємодією з базою даних тощо. Так як після компіляції Kotlin Multiplatform перетворюється у нативний код для відповідної операційної системи, тобто для iOS - це Swift [2], у той час як для Android OS - Байт-код [2], час виконання виконуваної бізнес-логіки є аналогічним до часу, який отримується при написанні її для двох платформ повністю окремо, а ось часу для самого написання, буде витрачено, майже у 2 рази менше. Тобто Kotlin Multiplatform являється однією з найбільш перспективною технологією побудови кросплатформених додатків на сьогодні.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Інтернет-протал статистичних даних Datareportal. URL:<https://datareportal.com/global-digital-overview> (дата звернення: 02.03.2021).
2. Офіційна документація мови програмування Kotlin, частина про Kotlin Multiplatform та його структуру. URL:<https://kotlinlang.org/docs/multiplatform.html> (дата звернення: 02.03.2021).
3. Офіційна документація Kotlin Multiplatform. URL:<https://kotlinlang.org/docs/mobile/home.html> (дата звернення: 02.03.2021).
4. Офіційна документація бібліотеки Ktor. URL:<https://ktor.io/docs/welcome.html> (дата звернення: 02.03.2021).
5. Офіційна документація бібліотеки Kodein. URL:<https://docs.kodein.org/kodein-di/7.3/index.html> (дата звернення: 02.03.2021).
6. Офіційна сторінка Flutter. URL:<https://flutter.dev/> (дата звернення: 02.03.2021).
7. Офіційна сторінка React Native. URL:<https://reactnative.dev/> (дата звернення: 02.03.2021).

**Урдзік В'ячеслав Віталійович** — студент групи 2ПІ-20б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: [urdziksv@gmail.com](mailto:urdziksv@gmail.com)

**Кателініков Денис Іванович**, кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [fuzzy2dik@gmail.com](mailto:fuzzy2dik@gmail.com).

**Urdzik Viacheslav V.** – student gr. 2PI-20b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [urdziksv@gmail.com](mailto:urdziksv@gmail.com)

**Katielnikov Denys Ivanovych**, PhD, Associate Professor of Software Engineering Department, Vinnytsia National Technical University, Vinnytsia, E-mail: [fuzzy2dik@gmail.com](mailto:fuzzy2dik@gmail.com).