

## **ВИКОРИСТАННЯ JETPACK COMPOSE У СУЧАСНІЙ МОБІЛЬНІЙ РОЗРОБЦІ ANDROID**

Бурим М. В.

Науковий керівник – к.т.н., доц. Імангулова З. А.

Харківський національний університет радіоелектроніки, каф. СТ  
м. Харків, Україна

e-mail: [maksym.burym@nure.ua](mailto:maksym.burym@nure.ua)

The work discusses the benefits of using Jetpack Compose library as the main toolkit for Android app development. It proves how Jetpack Compose simplifies animation development by providing a declarative approach, built-in animation support, automatic state management, and providing seamless integration into existing Android projects. This report would be valuable for the developers who want to create animation-rich interactive Android applications in short time.

У сучасному світі використання мобільних технологій стає все більш популярним, оскільки зараз майже половина населення Землі володіє смартфонами. Разом з цим ринок мобільних застосунків бурхливо розвивається, пропонуючи різноманітні рішення, що задовольняють будь-які потреби цільової аудиторії.

Це призвело до підвищення планки вимог до самих продуктів, до того ж кожен бізнес має на меті отримати на виході якісний продукт у короткі строки, скоротивши при цьому кінцеві витрати. Одним із засобів, що дозволяє втілити дані потреби при розробці Android-рішень, є бібліотека Jetpack Compose.

Jetpack Compose є сучасним інструментарієм розробки декларативного інтерфейсу користувача для додатків Android, представлений компанією Google як майбутнє побудови користувацьких інтерфейсів. Він пропонує ряд переваг та покращень у порівнянні з традиційним підходом на основі View, що робить його привабливим вибором для Android розробників [1].

Розглянемо основні причини, чому Jetpack Compose варто розглянути як основний інструментарій для розробки користувацьких інтерфейсів.

1. Декларативний та реактивний інтерфейс. Jetpack Compose застосовує декларативний підхід у побудові користувацьких інтерфейсів. Замість імперативної маніпуляції окремими view, розробник декларує який стан має мати користувацький інтерфейс на основі наявних даних програми. Таким чином, Compose автоматично оновлює інтерфейс кожного разу, коли змінюються дані програми, усуваючи при цьому потребу ручного виклику методів, що оновлюють весь екран цілком, і зменшуючи кількість шаблонного коду (бойлерплейту). Такий підхід дозволяє писати більш чистий та зручний для підтримки код, а також

підвищує можливість повторного використання наявного коду, скорочуючи тим самим час розробки і підвищуючи продуктивність розробників.

2. Розширена підтримка анімацій. Jetpack Compose має багато наявних видів анімацій з-під коробки, як, наприклад, зміна розміру, переміщення в площині, зміна прозорості тощо, а також інструментарій для створення власних. Це дозволяє розробникам легко створювати анімаційні ефекти без необхідності вручну визначати анімаційні властивості або використовувати сторонні бібліотеки для роботи з анімаціями.

3. Функція попереднього перегляду. Однією зі значущих особливостей Jetpack Compose є можливість миттєво бачити зміни у розроблюваному інтерфейсі за допомогою функції попереднього перегляду (Preview). Preview дозволяє розробникам візуалізувати компоненти користувацького інтерфейсу в режимі реального часу без необхідності розгортання програми на реальному пристрої або емуляторі.

Ця функція суттєво прискорює цикл розробки, оскільки розробники можуть швидко ітерувати дизайн інтерфейсу, експериментуючи при цьому з різними макетами, темами та станами і одразу ж бачити результат. Швидший цикл зворотного зв'язку підвищує продуктивність розробників і робить процес розробки більш динамічним та ефективним.

4. Керування станом компонентів з-під коробки. Керування станами є надзвичайно вагомим аспектом при побудові складних користувацьких інтерфейсів. Jetpack Compose пропонує вбудовані можливості керування станами, які спрощують обробку та оновлення стану інтерфейсу. Для цього Compose використовує елементи StateFlow та LiveData, які має мова програмування Kotlin, для керування та спостереження за змінами стану компонентів інтерфейсу. Даний механізм управління станами усуває необхідність у зовнішніх бібліотеках, або необхідність писати код вручну для обробки станів.

5. Безшовна інтеграція з уже існуючими проектами на Android. Jetpack Compose спроектований з урахуванням наявної кодової бази на основі XML, тому розробники мають можливість впроваджувати Compose поступово, додаючи компоненти Compose до вже існуючих екранів. Така гнучкість дозволяє розробникам використовувати можливості Compose у певних частинах програми, поступово переводячи модуль користувацького інтерфейсу на Compose.

Android є найбільш популярною мобільною платформою у світі. Її широкі можливості роблять її привабливим вибором для розробки мобільних додатків, а використання Android у поєднанні з Jetpack Compose відкриває нові можливості для створення швидких, гнучких та красивих додатків з високим рівнем користувацького досвіду в короткі терміни.

Список використаних джерел:

1. Basics of Jetpack Compose in Android. URL: <https://www.geeksforgeeks.org/basics-of-jetpack-compose-in-android/> (дата звернення: 04.03.2024).