

УДК 519.246.8(075.8)

ІННОВАЦІЙНІ РІШЕННЯ ДЛЯ СТВОРЕННЯ КРОСПЛАТФОРМЕННИХ МОБІЛЬНИХ ДОДАТКІВ

О.О. Гомілко, магістрант

Київський національний університет технологій та дизайну

Т.І. Демківська, кандидат технічних наук, доцент

Київський національний університет технологій та дизайну

Ключові слова: мобільна розробка, кросплатформеність, Android, iOS, flutter, dart.

Головною метою є дослідження технологій Flutter та Dart для розробки кросплатформенних мобільних додатків. Акцент зроблено на наступних моментах:

- дослідження архітектури Flutter;
- дослідження можливості написання одного програмного коду одночасно для Android та iOS платформ;
- дослідження можливості розробки нативних додатків для Android та iOS платформ;
- дослідження «сильних» сторін розробки;
- порівняння з існуючими аналогами для отримання порівняльної характеристики.

Flutter – це безкоштовний фреймворк з відкритим кодом, що був створений компанією Google та випущений у травні 2017 року як аналог ReactNative від Facebook. У кількох словах, цей фреймворк дозволяє створювати нативні мобільні застосунки на основі лише однієї бази коду. Це означає, що ви можете використовувати одну мову програмування та одну кодову базу для створення двох різних додатків (для iOS та Android). Flutter складається з двох важливих частин:

- SDK (Software Development Kit): набір інструментів, які допоможуть вам розробити ваші програми. Сюди входять інструменти для компіляції вашого коду в нативний машинного коду (код для iOS та Android).
- Framework (UI бібліотека на основі віджетів): набір елементів інтерфейсу користувачів (кнопки, поля введення тексту, повзунки тощо), які ви можете персоналізувати для власних потреб.

Основна ідея побудови UI додатку використовуючи Flutter – це побудова інтерфейсу за допомогою написання коду. Ви завжди будете дерево з віджетів у вашому застосунку. У вас не буде drag-and-drop інтерфейсу для додавання кнопок чи тексту на екран, який бачить юзер, натомість ви будете писати лише код.

Віджетом називається абсолютно кожний елемент у Flutter. Не важливо, чи це текст, кнопка, іконка або ж навіть поле для введення тексту – усі ці елементи є віджетами (рис. 1). Розглянемо приклад:

Верхня панель – це віджет, що містить у собі інші менші віджети (дерево віджетів). Назви, поля для вводу, поле для прикріплення документів, кнопка для відправлення – усе це віджети. Абсолютно весь додаток буде побудований з віджетів, навіть уся сторінка є віджетом, та і весь додаток «загорнутий» у віджет. Що ж таке віджет? Це шматок коду, написаний розробником, що виконує певну інструкцію, щоби відобразити потрібний елемент на екрані користувача.

Кожен об'єкт є віджетом!

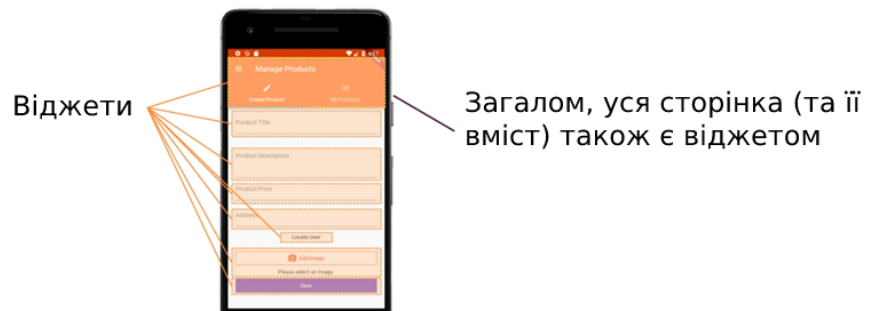


Рисунок 1 – Віджети у Flutter

Використовуючи віджети, ми будемо так зване дерево із віджетів, де коренем дерева є наш додаток, а його сини – це відповідно інші віджети, які будуть відповідати за те, щоб відобразити необхідну нам сторінку.

Застосунки, що створюються на основі Flutter використовуються мову програмування Dart. Мова була створена компанією Google у жовтні 2011 року. Dart – це об'єктно орієнтована, строго типізована мова програмування.

Dart зосереджується на розробці інтерфейсу, і ви можете використовувати його для створення мобільних застосунків та вебдодатків.

Код, написаний на Dart, використовує Flutter фреймворк – це набір віджетів (вбудованих у Flutter, а також ваші власні), що необхідно скомпілювати для додатків на Android та iOS. Flutter компілює Dart код у нативний код для кожної із цих платформ за допомогою FlutterSDK. Як результат ви отримаєте додаток для кожної з платформ на основі вашого коду. Flutter не використовує платформенні примітиви. Наприклад, вам необхідно додати кнопку. Це не означає, що при компіляції Flutter створює нативний еквівалент кнопки для Android та iOS, натомість Flutter має власний механізм, що дозволяє контролювати на рендерити кожен піксель на екрані, що відображається користувачеві. Це надає Flutter повний контроль над інтерфейсом.

Список використаних джерел

1. Design Patterns: Elements of Reusable Object-Oriented Software [Книга]: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides – 1994.
2. Clean Architecture: A Craftsman's Guide to Software Structure and Design [Книга]: Robert C. Martin – 2017.