

ОСОБЛИВОСТІ РОЗРОБКИ ANDROID ДОДАТКІВ. ПОРІВНЯННЯ РІЗНИХ СПОСОБІВ РОЗРОБКИ

Соколова В. К.

Науковий керівник – к.т.н., ст. викл. Ткачов В. М.

Харківський національний університет радіоелектроніки
(61166, Харків, просп. Науки, 14, каф. ЕОМ, тел. (057) 702-13-54)

e-mail: des@nure.ua

As practice has shown, this is an actual dilemma for both customers and novice developers who want to acquire the most useful experience for their future career. Programming in a native environment requires knowledge of the relevant languages. For Android, this is Kotlin and / or Java, and for iOS, Swift and / or Objective-C. In principle, you can get by with one of two for each platform, especially since Google is actively developing Kotlin, and Apple is investing a lot of effort in improving Swift. This publication offers an overview of a comparison of different approaches to developing android applications.

Кожен день кількість мобільних користувачів зростає, а отже і розвивається світовий ринок мобільних додатків. Існує два підходи розробки: нативний та крос-платформний.

Нативний підхід

Додаток розробляється на "рідній" мові для кожної платформи. Для Android - це Kotlin або Java, а для iOS - це Swift або Objective-C. Як правило можна обійтися одним з двох для кожної платформи, тим більше, що Google активно розвиває Kotlin, а Apple вкладає великі зусилля у вдосконалення Swift.

Переваги використання нативних мов в тому, що розроблений на них додаток буде працювати швидше та зможе використовувати всі можливості платформи, а інтерфейс буде зрозумілий та зручний будь-якому користувачеві платформи. До того ж розробляти нативні додатки дуже просто. Якщо від програми потрібно отримати максимальну продуктивність, то використовують нативну розробку. Головний мінус цього підходу в тому, що під кожен платформу доводиться створювати окремий додаток, хоча більша частина функціональних можливостей буде однаковою. Якщо ж потрібно розробити кілька додатків, то цей процес буде більш тривалим і витратним.

Кросплатформна розробка

Існує два основних способи розробити кросплатформний додаток: зробити це вручну, написавши код на C++ та обгортки для різних платформ, або використовувати одну із спеціально розроблених технологій.

Розробка вручну.

Перший спосіб полягає у тому, що код на C++ можна запустити будь-де. В Android для цього використовується NDK, Windows Phone - Managed C, інші платформи також мають свої способи обробити і запустити код.

Інша справа - такий код буде обмежений у своїх можливостях. Наприклад, в Android він не зможе звернутися до екрану або навіть самостійно запуститися. Щоб уникнути цих обмежень, спочатку пишеться бібліотека з основною логікою на C ++, а потім - обгортка на нативною мовою, яка запускає бібліотеку та забезпечує її взаємодію з пристроєм.

Другий спосіб полягає у використанні однієї з технологій кросплатформної розробки, яких на сьогоднішній день існує чимало. Існує популярний кросплатформний фреймворк - Flutter. Для роботи з ним потрібно знати мову програмування Dart. Flutter стає все більш популярним серед програмістів (розробників мобільних версій eBay, Aliexpress і навіть Meduza.io). Основні недоліки кросплатформної розробки - складність розробки функціональності та збільшення витрат. Кросплатформні додатки нерідко працюють повільніше нативних.

Найголовнішою відмінністю нативного підходу від кросплатформного полягає у тому, що один і той же додаток розробляється окремо і під iOS, і під Android, а в кросплатформному підході розробляється лише за один раз. Для того, щоб зрозуміти, яким підхід краще, потрібно оцінити складність та унікальність додатку. Прості рішення вигідніше розробляти на кросплатформних технологіях, але чим складніше функціональність - тим більше вигідною стає нативна розробка.

Список використаної літератури:

1. Vitalii Tkachov, Anna Budko, Kateryna Hvozdetska and Daryna Hrebenuk. Method of Building Dynamic Multi-hop VPN Chains for Ensuring Security of Terminal Access Systems // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T): Kharkiv 06-09 oct. 2020, Kharkiv.

2. Tkachov, V., Bondarenko, M., Ulyanov, O., & Reznichenko, O. (2019, December). Overlay Network Infrastructure for Remote Control of Radio Astronomy Observatory. In 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT) (pp. 161-165).

3. Tkachov, V., Hunko, M., Volotka, V.: Scenarios for Implementation of Nested Virtualization Technology in Task of Improving Cloud Firewall Fault Tolerance. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), pp. 759-763. IEEE (2019).

4. Hunko M.A., Tkachov V.M. Development of a module for sorting the ipaddresses of user nodes in cloud firewall protection of web resources. Дев'ята міжнародна науково-технічна конференція «Сучасні напрями розвитку інформаційнокомунікаційних технологій та засобів управління». 2019. С. 30.