

ОГЛЯД КРОСПЛАТФОРМЕННИХ ФРЕЙМВОРКІВ ДЛЯ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ

На даний момент швидко розвивається кросплатформний підхід, який дозволяє розробляти додатки для різних платформ. Переваги при створенні кросплатформних додатків є:

- суттєве зменшення витрат на адаптацію існуючого програмного забезпечення або розробку нового;
- висока швидкість роботи завдяки компіляції JavaScript в нативний код;
- зменшення кількості фахівцем при створенні мобільного додатку під 2-3 мобільні операційні системи на кросплатформному фреймворку;
- можливість інтеграції нативних модулів.

Найбільш популярними такими фреймворками є: React Native, Xamarin, Flutter тощо.

Розглянемо детальніше такі фреймворки як React Native та Flutter, як найбільш популярні на даний час.

Фреймворк React Native. React Native не використовує HTML. Натомість повідомлення з потоку JavaScript використовуються для маніпулювання нативними View.

Принцип роботи React Native дуже схожий з React, за винятком одного важливого моменту—React Native не маніпулює DOM за допомогою Virtual DOM, тому що в мобільних додатках він відсутній. Virtual DOM працює у фоновому процесі, який інтерпретує JavaScript, що написаний безпосередньо мобільними розробниками, на кінцевому пристрої і спілкується із нативною кодовою базою за допомогою серіалізації, пакетного та асинхронного мостів.

Найвигідніший код обертається за допомогою реактивних компонентів, які взаємодіють з нативними application programming interface (API) за допомогою декларативної парадигми JavaScript та інтерфейсу користувача. Це надає можливість розробляти нативні програми за допомогою невеликих команд розробників, а в деяких випадках можна обмежитися одним розробником.

Фреймворк Flutter. Flutter представляє фреймворк від компанії Google, який дозволяє створювати кросплатформні програми, які можуть використовувати один і той же код. Спектр платформ, в яких він застосовується, широкий – це веб-додатки, мобільні програми під Android та iOS, графічні програми під настільні операційні системи Windows, MacOS, Linux тощо.

Дебютна версія Flutter, яка працювала на операційній системі Android, була відома під кодовою назвою «Sky». SDK був висвітлений на саміті розробників Dart у 2015 року, із заявкою на можливість послідовно відображати зображення з частотою більш ніж 120 кадрів в секунду. Під час виступу розробників з Google у Шанхаї було оголошено Flutter Release Preview 2 –потужний реліз перед стабільним релізом Flutter 1.0. На заході Flutter Live, 4 грудня 2018 року, було оголошено о Flutter 1.0. На заході Flutter Interactive, 11 грудня 2019 року, з'явився Flutter 1.12 та було оголошено, що Flutter – це перша плат-форма інтерфейсу, що призначена для обчислення навколишнього середовища.

Для створення додатків Flutter використовується мова Dart, яка містить багато вдосконалень. Під час розробки програм, Flutter транслює код написаний на Dart у нативний код програми за допомогою Dart AOT (компіляція програми перед його запуском), яку можна за-пускати на Android або iOS або іншій платформі. Однак при розробці програми для її прискорення Flutter використовує JIT (компіляція програми під час його запуску).

Що використовується для розробки під Flutter? Для написання коду програми можна використовувати будь-який улюблений текстовий редактор і потім за допомогою утиліт командного рядка з Flutter SDK компілювати програму. Однак для таких середовищ, як Android Studio та IntelliJ IDEA, а також текстового редактора Visual Studio Code, компанія Google випустила спеціальні плагіни, які дозволяють спростити розробку. Тому найчастіше для розробки під Flutter використовуються саме Android Studio та IntelliJ IDEA.

Таким чином, набір інструментів представлених фреймворків React Native та Flutter надають можливість створення, інтеграції і впровадження продуктів із використанням однієї мови програмування для кросплатформних мобільних додатків у багатьох сферах: в медицині і охороні здоров'я, для навчання студентів, впровадження технологій читання по губах, миттєвих перекладачів, для моделювання та проектування у виробництві, комерційних цілях, використання машинного зору для запровадження технологій самокеруючих автомобілів, для підвищення показників продажів і спілкування з споживачами, налагодження алгоритму продажів тощо.