**Нативные, гибридные и кроссплатформенные приложения мобильных устройств: понятие, примеры, технология создания.**

**Нативные приложения**

Понятие: Нативные приложения создаются для конкретной платформы (iOS или Android) с использованием инструментов и языков программирования, предназначенных для этой платформы.

Примеры:

iOS: Instagram, WhatsApp, Spotify (написаны на Swift или Objective-C).

Android: Google Maps, Facebook, TikTok (написаны на Kotlin или Java).

**Технология создания:**

Языки программирования: Swift, Objective-C (iOS); Kotlin, Java (Android).

Инструменты разработки: Xcode (iOS), Android Studio (Android).

Преимущества: Высокая производительность, доступ ко всем возможностям устройства и платформы, более качественный пользовательский интерфейс.

Недостатки: Требуется отдельная разработка и поддержка для каждой платформы, что увеличивает затраты времени и ресурсов.

**Гибридные приложения**

Понятие: Гибридные приложения объединяют элементы нативных и веб-приложений. Они разрабатываются с использованием веб-технологий (HTML, CSS, JavaScript) и оборачиваются в нативный контейнер для запуска на мобильных устройствах.

Примеры:

Twitter, Uber, Evernote (разработаны с использованием гибридных фреймворков).

**Технология создания:**

Языки программирования: HTML, CSS, JavaScript.

Фреймворки: Apache Cordova, Ionic, PhoneGap.

Преимущества: Быстрая разработка, возможность кроссплатформенной поддержки с единой кодовой базой, снижение затрат.

Недостатки: Ограниченная производительность, доступ к нативным функциям может быть ограничен, менее качественный пользовательский интерфейс.

**Кроссплатформенные приложения**

Понятие: Кроссплатформенные приложения разрабатываются с использованием инструментов, которые позволяют создавать приложения для нескольких платформ (iOS и Android) из одной кодовой базы.

Примеры:

Airbnb, Alibaba, Skype (разработаны с использованием кроссплатформенных фреймворков).

Технология создания:

Языки программирования: Dart, C#, JavaScript.

Фреймворки: Flutter, React Native, Xamarin.

Преимущества: Единая кодовая база для всех платформ, сокращение времени разработки, снижение затрат.

Недостатки: Возможны ограничения в доступе к нативным функциям, производительность ниже, чем у нативных приложений.

\_\_\_\_\_

Нативные, гибридные и кроссплатформенные приложения мобильных устройств являются технологиями разработки программного обеспечения для мобильных устройств. Они отличаются друг от друга в терминах используемых языков программирования, доступности для разных платформ и производительности.

**Нативные приложения:**

Понятие: Нативные приложения разрабатываются специально для определенной платформы (например, iOS или Android), используя собственные языки программирования и инструменты разработки, предоставленные производителем платформы.

Примеры: Instagram для iOS, Facebook для Android.

Технология создания: Для разработки нативных приложений на iOS используется язык программирования Swift или Objective-C, а для Android - Java или Kotlin.

**Гибридные приложения:**

Понятие: Гибридные приложения создаются с использованием веб-технологий, таких как HTML, CSS и JavaScript, и запускаются внутри специальной обертки, которая позволяет им работать на разных платформах.

Примеры: Airbnb, Uber.

Технология создания: Разработка гибридных приложений может осуществляться с использованием различных фреймворков, таких как Apache Cordova (PhoneGap), React Native и Ionic.

**Кроссплатформенные приложения:**

Понятие: Кроссплатформенные приложения разрабатываются с использованием одной базовой кодовой базы и могут работать на разных платформах без изменений в исходном коде.

Примеры: WhatsApp, Slack.

Технология создания: Для создания кроссплатформенных приложений широко используются фреймворки, такие как React Native, Flutter и Xamarin.

\_\_\_\_\_

**Создать мобильное приложение для ввода данных о студенте (фамилия, имя, пол, группа, курс) с последующим сохранением в базу данных SQLite. Сохраненные данные в базе данных вывести на отдельной странице мобильного приложения.**