Введение в компьютерную науку

Mobile app development



Salymbekov University Miss Aliia Beishenalieva aliya.beiwenalieva@gmail.com

Recap

□ Что мы прошли на прошлом уроке?

- □ Что такое мобильное приложение?
 - Программы, созданные для работы на мобильных устройствах (смартфоны, планшеты).
 - Приложения можно установить через магазины приложений (App Store, Google Play).
 - Разделение на нативные, веб-приложения и кроссплатформенные.
 - Примеры: социальные сети, игры, утилиты и бизнес-приложения.

- □ Основные типы мобильных приложений
 - **Нативные приложения:** создаются для конкретной платформы (iOS, Android). Высокая производительность, доступ ко всем функциям устройства.
 - **Веб-приложения:** работают через браузер, не требуют установки, но имеют ограниченный доступ к системным ресурсам.
 - **Кроссплатформенные приложения:** один код для обеих платформ (React Native, Flutter), хорошая производительность, сокращение времени разработки.

- □ Основные платформы для разработки
 - iOS: разработка на языке Swift с использованием Xcode.
 - Android: разработка на языке Kotlin или Java с использованием Android Studio.
 - **Кроссплатформенные инструменты:** React Native (JavaScript), Flutter (Dart).

- □ Основные языки и инструменты
 - Языки для нативной разработки:
 - iOS: Swift, Objective-C.
 - Android: Kotlin, Java.
 - Языки для кроссплатформенной разработки:
 - React Native: JavaScript.
 - Flutter: Dart.
 - Инструменты: Xcode, Android Studio, Visual Studio Code.

iOS vs Android

- ❖ Нативная разработка для iOS
- ≽Язык Swift.
- ➤Интегрированная среда разработки (IDE): Xcode.
- ▶Структура приложения: контроллеры, вьюшки, модели (MVC).
- Пример простого приложения: создание калькулятора.

- ❖ Нативная разработка для Android
- ≻Язык Kotlin (или Java).
- ➤ Android Studio как основная среда разработки.
- ▶Архитектура приложений: MVVM.
- ▶Пример простого приложения: ТоОо список.

React Native vs Flutter

- ❖ Разработка на React Native
 - Использование JavaScript.
 - > Компоненты для создания UI.
 - Пример приложения:приложение для списка задач.
 - Одновременная разработка для iOS и Android.

- ❖ Разработка на Flutter
 - Язык программирования Dart.
 - Виджеты как основа интерфейса.
 - Пример приложения: калькулятор.
 - Высокая производительность близка к нативной.

❖ UI/UX для мобильных приложений

- •**UI (Пользовательский интерфейс):** как выглядит приложение.
- •UX (Пользовательский опыт): как пользователь взаимодействует с приложением.

Принципы дизайна: простота, интуитивность, доступность.

Инструменты для дизайна: Figma, Adobe XD.

Работа с сервером и данными

- Подключение к API (REST, GraphQL).
- Работа с JSON данными.
- Локальное хранение данных: SQLite, Core Data.
- Пример: получение данных о погоде через API.

□ Публикация приложений в App Store и Google Play

- 1. Процесс публикации: регистрация разработчика, подготовка и загрузка приложения.
- 2. Требования к приложениям: политика безопасности, соответствие стандартам платформы.
- 3. Монетизация: реклама, встроенные покупки, подписки.

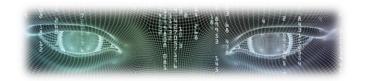
- Вопросы для анализа одного существующего приложения:
- Насколько удобно приложение в использовании? Есть ли сложности в навигации?
- Какие сильные стороны дизайна (цвета, типография, расположение элементов)?
- Какие улучшения можно предложить для улучшения пользовательского опыта (UX)?

HW

- Научиться работать с инструментом для дизайна Figma
 - Сделать один дизайн для мобильного приложения

Thanks for your attention

Mobile app development



Salymbekov University Miss Aliia Beishenalieva aliya.beiwenalieva@gmail.com