Лекция 2.Архитектура Android Studio.Структура операционной системы Android. Работа с Dalvik/ART (Android Runtime).

Цель лекции:

1. Ознакомить слушателей с архитектурой Android Studio, включая ее основные компоненты и инструменты.

2. Объяснить структуру операционной системы Android и её ключевые уровни: ядро Linux, системные библиотеки, фреймворк приложений и пользовательские приложения.

3. Рассмотреть работу Dalvik Virtual Machine (DVM) и причины перехода на Android Runtime (ART).

4. Показать различия между Dalvik и ART, их влияние на производительность приложений.

5. Продемонстрировать, как компоненты Android Studio и Android OS влияют на процесс разработки и оптимизацию мобильных приложений.

6. Дать студентам общее представление о том, как работают Android-приложения на уровне системной архитектуры и среды выполнения.

В лекции будут рассмотрены архитектура Android Studio и ее ключевые компоненты, включая редактор кода, систему сборки Gradle, эмулятор и инструменты отладки. Также будет обсуждена структура операционной системы Android: ядро Linux, системные библиотеки, фреймворк приложений и уровень пользовательских приложений. Особое внимание будет уделено Dalvik Virtual Machine (DVM) и Android Runtime (ART), их различиям, а также влиянию ART на производительность и оптимизацию работы приложений.

**Архитектура Android Studio** представляет собой сложную многослойную структуру, которая предоставляет разработчикам средства для эффективной разработки Android-приложений. Она включает различные компоненты и инструменты, которые помогают создавать, тестировать и развертывать приложения для мобильных устройств на платформе Android.

Android Studio построен на базе платформы IntelliJ IDEA, которая уже долгое время используется для разработки программного обеспечения на языке Java и других языках. IntelliJ предоставляет Android Studio мощный редактор кода с множеством возможностей, включая автозаполнение, анализ кода и поддержку различных плагинов. Однако, Android Studio включает дополнительные специфические для Android инструменты и функции, которые делают его специализированной IDE для Android-разработки.

Одним из ключевых компонентов архитектуры Android Studio является система сборки Gradle. Gradle отвечает за управление зависимостями проекта, сборку APK-файлов, их подпись и развертывание. Gradle очень гибок и позволяет разработчикам настраивать процесс сборки под свои нужды, будь то работа с различными конфигурациями для разных типов устройств или управление многими библиотеками и модулями проекта.

Также важным элементом архитектуры Android Studio является система управления проектом, которая строится вокруг концепции модульности. Каждый Android-проект делится на модули, и каждый модуль может представлять отдельное приложение, библиотеку или тестовый модуль. Это дает разработчикам гибкость в организации проектов, что особенно важно при разработке крупных приложений.

Еще один важный аспект архитектуры Android Studio – это Android SDK, который представляет собой набор инструментов и библиотек, предоставляющих API для разработки приложений. SDK включает в себя библиотеки, которые позволяют взаимодействовать с функционалом устройства (таким как камера, сенсоры, GPS), а также создавать интерфейсы пользователя. Android Studio поставляется с менеджером SDK, который позволяет легко обновлять и управлять версиями SDK, а также устанавливать различные пакеты для работы с конкретными устройствами и версиями Android.

**Эмулятор Android**, встроенный в Android Studio, играет значимую роль в архитектуре разработки. Он позволяет тестировать приложения на виртуальных устройствах, имитируя различные модели телефонов и планшетов с разными версиями Android. Этот эмулятор предоставляет возможность разработчикам проверять работу приложений на разных разрешениях экранов, устройствах с разными конфигурациями аппаратного обеспечения, а также симулировать разнообразные сценарии работы устройства, такие как разрядка батареи, слабый сигнал сети или режим энергосбережения.

**Инструменты анализа и отладки** также являются неотъемлемой частью архитектуры Android Studio. Profiler позволяет разработчикам мониторить использование ресурсов приложения, таких как память, процессор, батарея, и выявлять узкие места в производительности. В то время как инструмент логирования Logcat помогает отслеживать ошибки, предупреждения и другой вывод, полезный для диагностики проблем во время работы приложения.

**Интеграция с VCS (системой контроля версий), такой как Git**, позволяет разработчикам легко управлять версиями своего кода, работать в команде и эффективно отслеживать изменения. Android Studio поддерживает широкий спектр VCS, что делает процесс разработки более структурированным и упрощает совместную работу над проектом.

Поддержка различных плагинов и расширений также является ключевой особенностью архитектуры Android Studio. **Плагины** могут добавлять функциональность, автоматизировать процессы или предоставлять интеграции с другими инструментами, такими как платформы CI/CD или системы мониторинга ошибок. Разработчики могут создавать свои собственные плагины, чтобы настраивать среду разработки под специфические задачи.

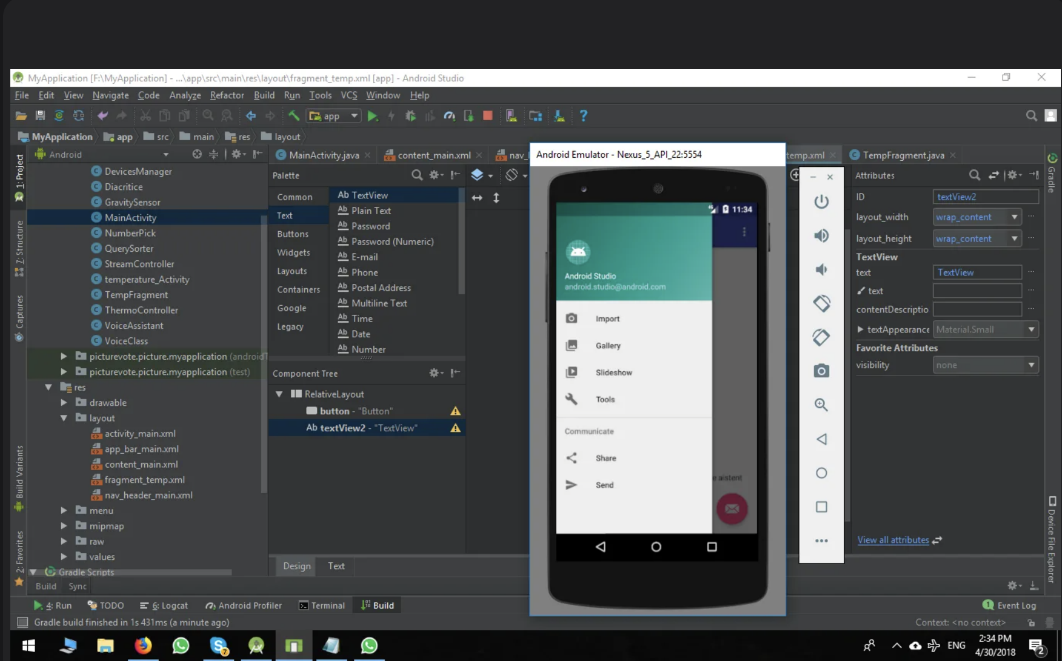


Рис.1 Архитектура Android Studio

Таким образом, Android Studio предлагает мощный и гибкий набор инструментов для разработки, отладки и развертывания Android-приложений. Его архитектура строится на основе многослойного взаимодействия инструментов, таких как IntelliJ, Gradle, Android SDK, эмулятор, инструменты анализа и отладки, а также системы управления версиями и модулей. Все эти компоненты работают вместе для обеспечения разработчикам эффективной и мощной среды для создания высококачественных Android-приложений.

Операционная система Android — это многослойная система, построенная на основе ядра Linux, которая включает различные уровни, каждый из которых отвечает за определенные функции. Основные слои структуры Android включают ядро, библиотеки, фреймворки, а также приложения, работающие на устройствах. Одной из ключевых особенностей Android является виртуальная машина для выполнения кода Java-приложений — Dalvik (в ранних версиях) и Android Runtime (ART) в более поздних версиях.

1. Ядро Linux

В основе операционной системы Android лежит модифицированное ядро Linux. Оно управляет аппаратными ресурсами устройства, такими как процессор, память, дисплей, камера, и обеспечивает безопасность и изоляцию процессов. Ядро также поддерживает сетевые функции, управление драйверами, файловую систему и взаимодействие с аппаратурой. Модификации ядра, сделанные для Android, адаптированы для мобильных устройств, обеспечивая энергосбережение, поддержку многозадачности и эффективную работу с ограниченными ресурсами.

2. Системные библиотеки

Второй уровень включает набор библиотек, реализованных на C и C++. Эти библиотеки предоставляют различные функции, необходимые для работы приложений и системы в целом. Например, библиотека `libc` отвечает за базовые системные вызовы, `liblog` обеспечивает логирование, а такие библиотеки, как OpenGL ES, предоставляют интерфейсы для работы с графикой.

Также в этот слой входят мультимедийные библиотеки (например, для работы с аудио- и видеоформатами), библиотеки для работы с базами данных (SQLite), и другие компоненты, которые обеспечивают выполнение важных функций для Android-приложений.

3. Android Runtime: Dalvik и ART

Наиболее важной частью операционной системы Android является среда выполнения приложений. До версии Android 5.0 (Lollipop) основным механизмом был Dalvik, а начиная с Android 5.0, его заменил Android Runtime (ART).

Dalvik Virtual Machine (DVM) — это виртуальная машина, разработанная специально для Android. Она оптимизирована для работы в условиях ограниченной памяти и вычислительных ресурсов мобильных устройств. Dalvik использует байт-код, который создается из скомпилированного Java-кода. Приложения компилируются в байт-код, который хранится в файлах формата `.dex` (Dalvik Executable). DVM отличается от стандартной Java Virtual Machine (JVM) тем, что она использует регистровую архитектуру, а не стековую, что более эффективно для мобильных устройств. Также Dalvik поддерживает многозадачность и параллельное выполнение приложений.

Однако у Dalvik были свои ограничения, такие как низкая производительность и относительно долгий процесс загрузки приложений, так как каждый раз байт-код интерпретировался заново.

С переходом на Android Runtime (ART) производительность Android была значительно улучшена. ART заменил Dalvik начиная с версии Android 5.0 и стал основной виртуальной машиной. ART использует концепцию предварительной компиляции (Ahead-of-Time, AOT), когда приложения компилируются в машинный код один раз при установке, а не каждый раз при запуске, как это было в Dalvik. Это сокращает время запуска приложений и повышает производительность.

Кроме того, ART включает механизм Just-In-Time (JIT) компиляции, который улучшает производительность во время выполнения программы за счет динамической компиляции частей кода. ART также предлагает более эффективное управление памятью, улучшенную отладку и сборку мусора (Garbage Collection), что способствует более плавной и быстрой работе приложений.

4. Фреймворк приложений

Фреймворк приложений — это слой, предоставляющий разработчикам инструменты и API для взаимодействия с системой. Он включает компоненты для работы с пользовательским интерфейсом (View System), управления жизненным циклом приложений (Activity Manager), взаимодействия с базами данных, сетями, геопозицией, уведомлениями и другими службами.

Этот уровень предоставляет такие важные компоненты, как:

- Activity Manager — управляет жизненным циклом приложений и обеспечивает многозадачность;

- Content Providers — отвечает за обмен данными между приложениями;

- Resource Manager — управляет ресурсами приложения, такими как изображения, локализованные строки и другие медиафайлы;

- Notification Manager — управляет уведомлениями, предоставляя пользователям информацию и предупреждения;

- Location Manager — предоставляет API для работы с геолокацией.

Этот уровень изолирован от системных компонентов и представляет собой высокоуровневые абстракции для упрощения разработки приложений.

5. Приложения

Самый верхний слой системы Android — это приложения. Сюда входят как стандартные приложения, такие как телефон, почта, браузер, так и пользовательские приложения, которые разработчики создают на языке Java или Kotlin с использованием Android SDK. Эти приложения взаимодействуют с пользователем и другими компонентами системы через интерфейсы, предоставляемые фреймворком приложений.

Таким образом, структура операционной системы Android является многослойной системой, начиная с ядра Linux, системных библиотек, Android Runtime (Dalvik/ART) и заканчивая фреймворком приложений и пользовательскими программами. ART представляет собой значительное улучшение по сравнению с Dalvik, обеспечивая лучшую производительность, более быструю загрузку приложений и эффективное управление памятью.

Видеоматреиалы:

(https://youtu.be/dNkotDjydiM?feature=shared, n.d.)

(https://youtu.be/IhH97tkAhs4?feature=shared, n.d.)

(https://youtu.be/1xW\_J74VfDo?feature=shared, n.d.)