|  |
| --- |
| **Page 1** |

**Лекция 2. ОБЗОР среды программирования**

Прежде чем начать разрабатывать приложения под Android, рассмотрим существующие

инструменты, которые подходят для этих целей. Можно выделить необходимые инструменты, без которых разработка мобильных приложений под Android просто невозможна. С другой стороны, существует большое количество вспомогательных систем, в какой-то степени упрощают процесс

разработки.

К обязательным инструментам относится Android SDK - набор средств

программирования, который содержит инструменты, необходимые для создания, компиляции и

сборки мобильного приложения.

Рассмотрим кратко наиболее важные инструменты, входящие в состав

Android SDK:

• **SDK Manager** - инструмент, позволяющий загрузить компоненты Android

SDK. Показывает пакеты Android SDK и их статус: установлен (Installed), не

установлено (Not Installed), доступные обновления (Update available).

• **Debug Monitor** - самостоятельный инструмент, предоставляющий графический интерфейс к

нескольких инструментов, предназначенных для анализа и отладки Android приложений:

o **DDMS (Dalvik Debug Monitor Server)** предоставляет услуги опрокидывания портов,

захвата экрана устройства, информацию о потоках и динамической памяти

устройства, вывода информации о действиях Android в реальном времени (logcat) и многие

другое.

o **Hierarchy Viewer**

позволяет налаживать и оптимизировать

пользовательский интерфейс Android программы.

• **Android Emulator (emulator)** - виртуальное мобильное устройство, которое

создается и работает на компьютере разработчика, используется для разработки и

тестирование мобильных приложений без привлечения реальных устройств.

• **AVD Manager** - предоставляет графический интерфейс для создания виртуальных

Android устройств (AVDs), предусмотренных Android Emulator, и управления ими.

• **Android Debug Bridge (adb)** - гибкий инструмент, позволяющий управлять

по состоянию эмулятора или реального Android устройства, подключенного к компьютеру.

Также может использоваться для установки Android программы (.apk файл) на

реальное устройство.

o Tracer for OpenGL ES - инструмент для анализа OpenGL | ES кода,

используемого в мобильном приложении, позволяет захватывать команды OpenGL |

ES и демонстрировать их по отдельным кадров, помогает понять как выполняются

графические команды.

Мы рассмотрели основные инструменты, входящие в состав Android SDK,

разумеется, не все и недостаточно подробно. Для более серьезного изучения

инструментов имеет смысл обратиться к сайту разработчиков (

http://developer.android.com/tools/help/index.html).

Для разработки мобильных приложений под Android уверенного владения

инструментами с SDK вполне достаточно. Если же возникают какие-то вопросы, дополнительные

инструкции по созданию проектов, компиляции, запуска из командной строки содержатся

в

руководстве

от

Google

(

http://developer.android.com/training/

basics / firstapp / index.html).

В современных условиях разработка ПО в большинстве случаев ведется с

использованием интегрированных сред разработки (IDE). IDE имеют несомненные

преимущества: процесс компиляции, сборки и запуска программы обычно автоматизирован, в

связи с чем для начинающего разработчика создать свое первое приложение труда не

представляет. Но чтобы заниматься разработкой всерьез, необходимо потратить силы и время

на изучение возможностей самой среды. Рассмотрим IDE, пригодные для

разработки под Android.

Для начала поговорим о среде разработки, которые рекомендует Google:

Android IDE (ADT) и Android Studio.

**Android IDE** - среда разработки под Android, основанная на Eclipse. предоставляет

интегрированные инструменты для разработки, сборки и отладки мобильных

приложений. В данном курсе Android IDE выбрана в качестве основной среды

разработки. Возможности этой среды более подробно рассмотрены в первой

лабораторной работе. Также там даны рекомендации по установке и

налаживанию среды, создания и запуска первого приложения как на эмуляторе,

так и на реальном устройстве.

**Android Studio** - среда разработки под Android, основанная на IntelliJ

IDEA. Подобно Android IDE, она предоставляет интегрированные инструменты для разработки и

отладки. Дополнительно ко всем возможностям, ожидаемым от IntelliJ, в Android

Studio реализованы:

• поддержка сборки приложения, основанной на Gradle;

• специфический для Android рефакторинг и быстрое исправление дефектов;

|  |
| --- |
| **Page 4** |

• lint инструменты для поиска проблем с производительностью, с юзабилити, с

совместимостью версий и других;

• возможности ProGuard (утилита для сокращения, оптимизации и обфускации

кода) и подписи приложений;

• основанные на шаблонах мастера для создания общих Android

конструкций и компонентов;

• WYSIWYG редактор, работающий на многих размерах экранов и разрешений,

окно предварительного просмотра, показывает запущенное приложение сразу на

нескольких устройствах и в реальном времени;

• встроенная поддержка облачной платформы Google.

Скачать последнюю версию Android Studio, а также получить рекомендации

по

установке,

настройке

и

начала

работы

можно

здесь:

http://developer.android.com/sdk/installing/studio.html.

**Рис.** **1.3.** Среда разработки Android Studio

Перейдем к рассмотрению других инструментов, пригодных для разработки

мобильных приложений под Android. Начнем с инструментов от Intel - Intel XDK и Intel

Beacon Mountain.

|  |
| --- |
| **Page 5** |

**Intel XDK** позволяет легко разрабатывать кроссплатформенные мобильные приложения;

включает в себя инструменты для создания, отладки и сборки ПО, а также

эмулятор устройств; поддерживает разработку для Android, Apple iOS, Microsoft Windows

8, Tizen; поддерживает языки разработки: HTML5 и JavaScript.

Последняя тема данного курса полностью посвящена изучению нового поколения

инструментальных средств разработки мобильных HTML5-приложений и Intel XDK,

предполагается разработка мобильного приложения с использованием этих инструментов.

**Intel Beacon Mountain** - среда разработки, позволяющая создавать

приложения для устройств, работающих под управлением ОС Android. предоставляет

инструменты необходимые для проектирования, разработки, отладки и оптимизации

приложений под Android. Освобождает разработчика от необходимости поддерживать систему

разработки в актуальном состоянии, следит за обновлениями и добавляет их в среду

разработки по мере появления. Поддерживает разработку для целевых платформ на основе

процессоров Intel Atom и ARM.

Beacon Mountain построена на основе Android IDE (Eclipse, Android ADT,

Android SDK), для более серьезной разработки и оптимизации добавлены следующие

инструменты Intel:

• **Intel \* Hardware Accelerated Execution Manager (Intel \* HAXM)** - аппаратно

поддерживаемый процессор виртуализации, использующая технологию виртуализации

Intel \* (Intel \* VT) для ускорения работы эмулятора в среде разработки.

• **Intel \* Graphics Performance Analyzers (Intel \* GPA) System Analyzer**

поддерживает мобильные устройства с процессором Intel Atom под управлением ОС Android.

Позволяет разработчикам оптимизировать загруженность системы при использовании

процедур OpenGL, предоставляя возможность получать множество системных метрик в

реальном времени, отражающие загруженность CPU, GPU и OpenGL ES API.

Разработчик может запустить несколько графических экспериментов для выявления узких

мест в обработке графики.

• **Intel \* Integrated Performance Primitives (Intel \* IPP) Preview** - библиотека

оптимизированной обработки данных и изображений, поддерживающая мобильные устройства с

платформой Intel под управлением ОС Android. Preview версия является частью полной

версии Intel IPP, которая тоже поддерживает ОС Android.

|  |
| --- |
| **Page 6** |

• **Intel \* Threading Building Blocks (Intel \* TBB)** - широко используемая,

признана библиотека шаблонов С ++ для создания масштабируемых приложений и

увеличение производительности. Поддерживает мобильные устройства с платформой Intel под

управлением Android. Проверенные методы позволяют разработчикам эффективно

распараллелить С ++ мобильные приложения, повышает производительность при снижении

энергетических затрат.

Нельзя обойти вниманием инструментарий Marmalade SDK.

**Marmalade SDK** - кроссплатформенное SDK от Ideaworks3D Limited. представляет

собой набор библиотек, образцов, инструментов и документации, необходимых для

разработки, тестирования и развертывания приложений для мобильных устройств.

Используется, в основном, для разработки игр. Многие получили признание игры,

такие как Cut The Rope и Plants vs. Zombies, были разработаны с использованием этого

программного средства. К сожалению, Marmalade SDK представляет собой проприетарное программное

обеспечения (самая дешевая лицензия $ 15 в месяц) и не может быть рекомендована в

данном учебном курсе, но читатель может самостоятельно попробовать бесплатную 30-

дневную версию, доступную по ссылке https://www.madewithmarmalade.com/free-trial.

**1.2 эмулятор**

**Эмуляция.** **Стандартный эмулятор Android**

**Эмуляция** (англ. *Emulation)* в вычислительной технике - комплекс программных,

аппаратных средств или их сочетание, предназначенное для копирования (или *эмулирование)*

функций одной вычислительной системы *(гостя)* на другой, отличной от первой,

вычислительной системе *(хост)* таким образом, чтобы эмулировать поведение можно

ближе соответствовало поведению оригинальной системы *(гостя).* Целью является максимально

точное воспроизведение поведения в отличие от различных форм компьютерного

моделирования, в которых имитируется поведение некоторой абстрактной модели (Википедия).

Эмулятор - виртуальное мобильное устройство, которое запускается на компьютере.

С помощью эмулятора можно разрабатывать и тестировать программы без использования

реальных устройств. На рисунке 1.4 приведен пример запущенного стандартного

эмулятора. Подробно работа с эмуляторами рассмотрена в лабораторной работе.

|  |
| --- |
| **Page 7** |

**Рис.** **1.4.** Эмулятор Android SDK в процессе запуска и приложение "Hello, world!"

К преимуществам использования эмуляторов можно отнести простоту их

использования и нулевую стоимость. Разработчику не нужно покупать огромную

количество устройств с различными характеристиками, чтобы проверить работоспособность

программы на разных смартфонах. Достаточно создать несколько эмуляторов с необходимыми

характеристиками и запустить на них приложение.

К сожалению, эмуляторы имеют и ряд недостатков:

• Требуют много системных ресурсов.

• Из-за различий в архитектуре процессоров компьютера и смартфона медленно

запускаются. Современные персональные компьютеры построены на архитектуре x86 и x64,

а большинство процессоров смартфонов на Android - ARM. Процесс эмуляции одной

архитектуры на другой чрезвычайно сложный и происходит довольно медленно.

• В некоторых случаях стандартного эмулятора недостаточно. Речь идет о

возможности смартфонов, которыми обычные компьютеры не обладают (например,

наличие датчика gps или акселерометра). В таких случаях полноценную

отладки можно провести только с использованием реального устройства.

**альтернативные эмуляторы**

Стандартный эмулятор, поставляемого вместе с Android SDK, не устраивает

многих. Существуют проекты, поддерживающие разработку и развитие альтернативных

эмуляторов. В качестве примера можно привести Genymotion - быстрый эмулятор Android

|  |
| --- |
| **Page 8** |

(по мнению его разработчиков). Он содержит предварительно настроенные образы Android

(x86 с аппаратным ускорением OpenGL). Genymotion доступен для Linux,

Windows и Mac OS X и требует для своей работы VirtualBox. Иными словами,

Genymotion представляет собой виртуальную машину с установленной ОС Android, которую

пользователь запускает так же, как и другие виртуальные машины. проблема высокого

потребление системных ресурсов, конечно, не исчезает, однако скорость запуска

существенно увеличивается.

**1.3 возможности налаживания НА реальных устройств**

Разработанное приложение можно запустить на реальном устройстве, например, на

смартфоне. Для этого необходимо проделать предварительную работу.

Для запуска приложений, разработанных в Android IDE, необходимо:

• Настроить устройство (включить режим отладки по USB).

• Настроить компьютер (Windows необходимо установить нужный

драйвер вручную, нужны права администратора).

• Настроить среду и запустить проект устройства.

Подробности отладки на реальных устройствах описаны в лабораторной

работе.