

Курс по Android разработке

Яборов Андрей Владимирович

avyaborov@gravity-group.ru

Марквирер Владлена Дмитриевна

vdmarkvirer@hse.ru

НИУ ВШЭ 2022



Цель и структура курса

Цель

Развить практические навыки разработки мобильных приложений под ОС Android

Структура курса

Вводное задание:

Работа со средой Android Studio

Baseline:

Типовые задачи (инд. проекты)

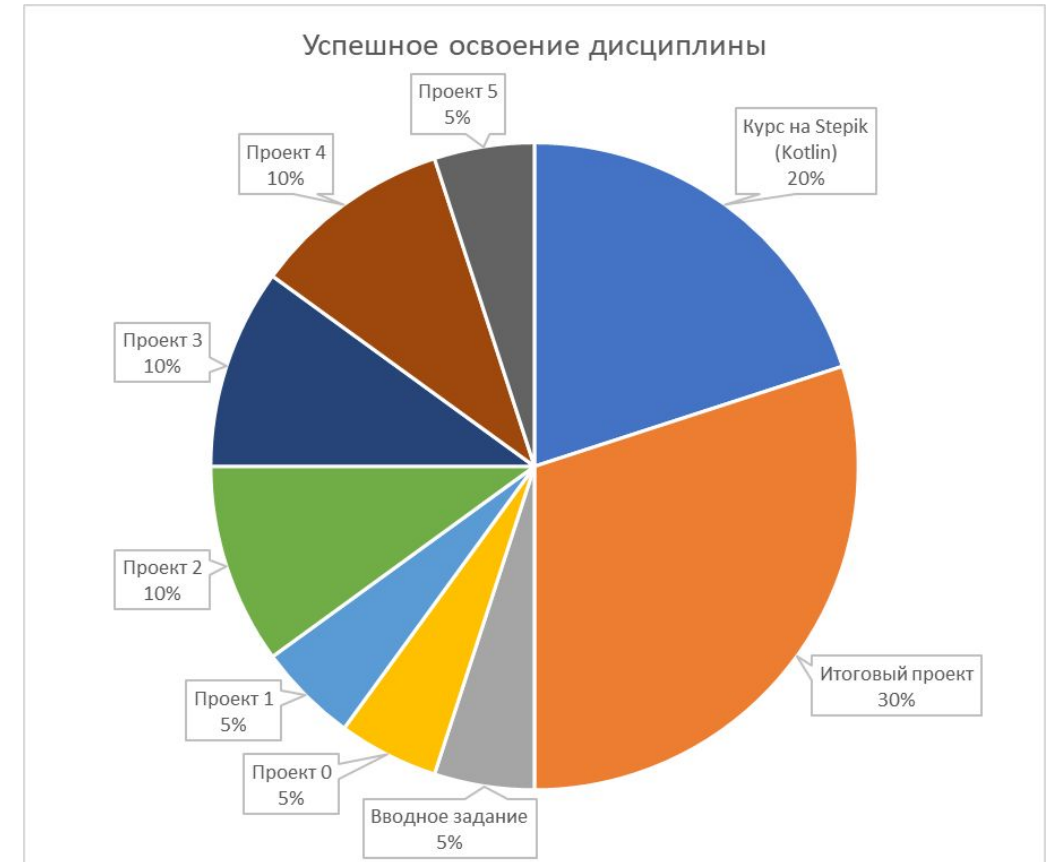
Итоговый проект:

Отчёт, приложение и презентация - формируется ср. оценка.

Проект групповой (3-4 человека)

Stepik:

<https://stepik.org/4792>



Разработка под Android

Урок 1

- Языки разработки. Краткий обзор Java, Kotlin. Fuchsia OS и Harmony OS
- Знакомство с IDE — Android Studio и системой сборки — Gradle
- Обзор основных возможностей IDE
- Эмуляторы, симуляторы, основные отличия
- Версии API
- Работа с дебагом и логирование
- Инструменты командной строки



Языки разработки. Краткий обзор Java



Строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems. Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Назван в честь марки кофе Java, которая, в свою очередь, получила наименование одноимённого острова (Ява), поэтому на официальной эмблеме языка изображена чашка с горячим кофе.

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik (начиная с Android 5.0 Lollipop виртуальная машина заменена на ART). Для такой компиляции используется дополнительный инструмент, а именно Android SDK (Software Development Kit), разработанный компанией Google.

Языки разработки. Краткий обзор Kotlin



Статически типизированный, объектно-ориентированный язык программирования, работающий поверх Java Virtual Machine и разрабатываемый компанией JetBrains. Язык назван в честь острова Котлин в Финском заливе, на котором расположен город Кронштадт

Авторы ставили целью создать язык более лаконичный и типобезопасный, чем Java, и более простой, чем Scala. Язык полностью совместим с Java, что позволяет java-разработчикам постепенно перейти к его использованию; в частности, в Android язык встраивается с помощью Gradle, что позволяет для существующего android-приложения внедрять новые функции на Kotlin без переписывания приложения целиком.

На Google I/O 2019 было объявлено, что язык программирования Kotlin стал приоритетным в разработке под Android

Fuchsia OS и Harmony OS

Fuchsia — операционная система, разрабатываемая корпорацией Google.

Впервые была обнаружена на GitHub в августе 2016 года без каких-либо официальных объявлений со стороны Google. В отличие от предыдущих развиваемых Google операционных систем, таких как Chrome OS и Android, которые основаны на ядре Linux, Fuchsia базируется на новом микроядре под названием «Zircon», производном от «Little Kernel», и является небольшой ОС, предназначенной для встраиваемых систем.

Пользовательский интерфейс и приложения Fuchsia написаны с помощью комплекта для разработки мобильных приложений Flutter, использующего язык Dart. Flutter также предлагает движок рендеринга на основе Vulkan под названием «Escher» с особой поддержкой «объёмных мягких теней», который Ars Technica описал как «специально созданный для работы с теневыми текстурами Material Design».

Благодаря комплекту разработки программного обеспечения Flutter, предлагающему кросс-платформенные возможности, пользователи могут устанавливать части Fuchsia на устройствах Android.

Изначально заявленная как тестовая платформа, предназначенная для испытания новых функций для уже существующих операционных систем, в конечном итоге она вышла в релиз, став вполне себе коммерческой. Google установила её на свои **умные дисплеи Nest Hub**, заменив созданную специально под них операционку Cast OS.

<https://fuchsia.dev/>

Fuchsia OS и Harmony OS

При желании Google может перевести Android, Chrome OS, Wear OS и другие ОС на Zircon так, чтобы пользователи этого даже не заметили. В результате изменится программная структура платформы, усилится её защищённость от внешних воздействий и возрастет стабильность, но сами пользователи не заметят никаких изменений. Fuchsia OS — это будущее операционных систем Google. Буквально недавно компания Samsung объявила информацию о возможном переходе на данную ОС.

HarmonyOS (ранее известная как *Hongmeng OS*, китайский: 鸿蒙系统; пиньинь: Hóngméng Xìtǒng) — операционная система на базе Android, разрабатываемая компанией Huawei. Полноценная разработка HarmonyOS началась в мае 2016 года.

Операционная система разработана для интеллектуальных устройств, таких как смарт-телевизоры, и использовалась в качестве мобильной операционной системы. Ожидается, что она станет бесплатной распределенной операционной системой на основе микроядра с открытым исходным кодом.

В апреле 2021 г. Huawei опубликовала официальное заявление, в котором говорится, что операционная система HarmonyOS не является заменой Android или iOS. По словам представителей Huawei, это совершенно новая операционная система. Huawei заявила, что HarmonyOS позиционируется как операционная система, созданная для эпохи Интернета Всего (Internet of Everything)

<https://www.harmonyos.com/en/>

Знакомство с IDE. Система сборки — Gradle

Gradle — система автоматической сборки, построенная на принципах Apache Ant и Apache Maven, но предоставляющая DSL на языках Groovy и Kotlin вместо традиционной XML-образной формы представления конфигурации проекта.

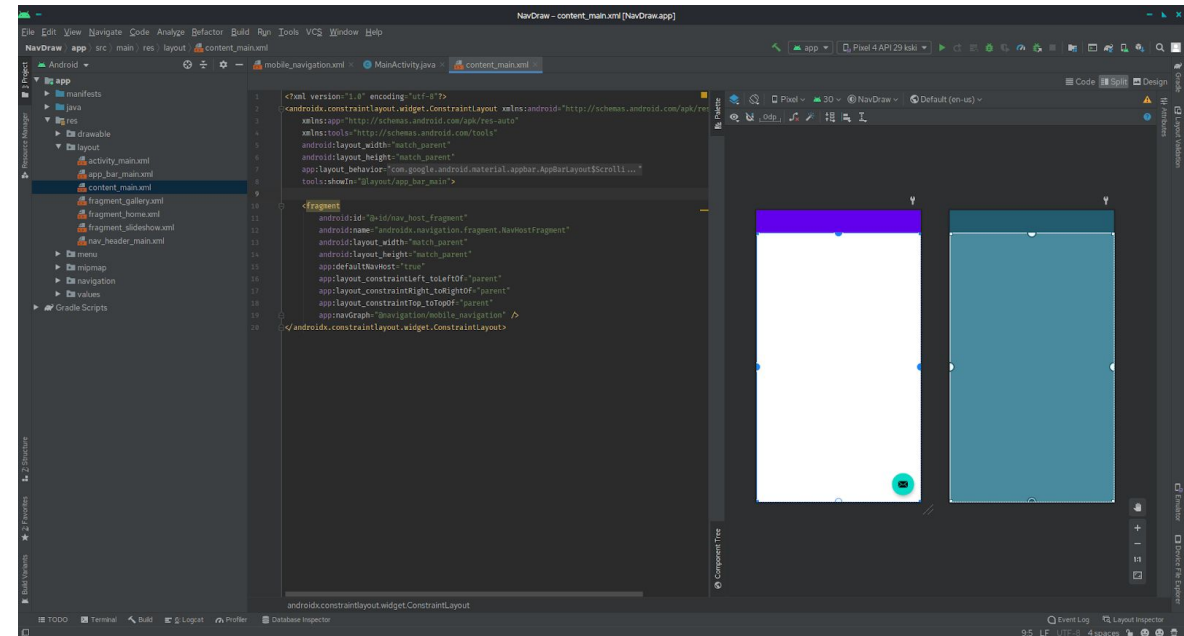
Gradle был разработан для расширяемых многопроектных сборок, и поддерживает инкрементальные сборки, определяя, какие компоненты дерева сборки не изменились и какие задачи, зависимые от этих частей, не требуют перезапуска.

Для приложений состоящих из модулей удобно использовать мультипроектный подход Gradle.

Каждый модуль расположен в своей директории. Каждый модуль может включать дочерние, которые будут расположены во вложенных директориях

Знакомство с IDE — Android Studio

Android Studio — интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O. Android Studio основана на программном обеспечении IntelliJ IDEA от компании JetBrains.



Обзор основных возможностей IDE

Новые функции появляются с каждой новой версией Android Studio. На данный момент доступны следующие функции:

- Расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана
- Сборка приложений, основанная на Gradle
- Различные виды сборок и генерация нескольких .apk файлов
- Рефакторинг кода
- Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое
- Встроенный ProGuard и утилита для подписывания приложений
- Шаблоны основных макетов и компонентов Android
- Поддержка разработки приложений для Android Wear и Android TV

Обзор основных возможностей IDE. ч2

Новые функции появляются с каждой новой версией Android Studio. На данный момент доступны следующие функции:

- Встроенная поддержка Google Cloud Platform, которая включает в себя интеграцию с сервисами Google Cloud Messaging и App Engine
- Начиная с версии Android Studio 2.1 способна работать с обновленным компилятором Jack, а также получила улучшенную поддержку Java 8 и усовершенствованную функцию Instant Run
- Начиная с Platform-tools 23.1.0 для Linux исключительно 64-разрядная
- В Android Studio 3.0 по стандарту включены инструменты языка Kotlin основанные на JetBrains IDE

Эмуляторы, симуляторы — Основные отличия

Эмуляция — это воспроизведение работы программы или системы (а не какой-то её мизерной части) с сохранением ключевых её свойств и принципов работы. **Эмуляция** выполняет программный код в привычной для этого кода среде, состоящей из тех же компонентов, что и эмулируемый объект.

Эмулируются все основные компоненты устройства, в том числе процессор, память и устройства ввода/вывода. Поскольку эмулируется и процессор, совершенно не обязательно чтобы в гостевой системе (эмулируемом устройстве) процессор был того же типа, что и в хост-системе (эмулирующем устройстве)

Симуляция — это воспроизведение работы программы-оригинала сугубо виртуально, на движке специальной программы (средство разработки курсов, к примеру). Симуляция лишь имитирует выполнение кода, а не копирует его, всё виртуально на 100%, всё понарошку.

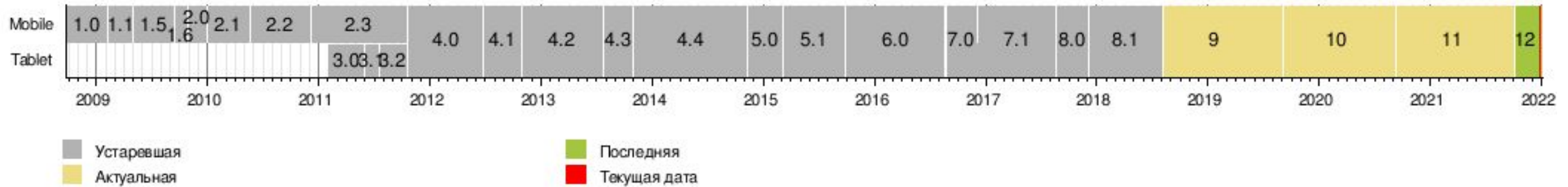
Эмуляторы, симуляторы — Основные отличия

Эмулятор ПО — полнофункциональный аналог оригинального ПО, либо его версия, в которой может быть предусмотрен ряд ограничений по функционалу, возможностям и поведению ПО

Симулятор ПО — модель оригинального ПО, в которой реализуется логика работы этого ПО (частично или полностью), имитируется поведение ПО, копируется его интерфейс

Версии API

Обновления версий Android



С момента выхода первой версии в сентябре 2008 года произошло 40 обновлений системы. Эти обновления, как правило, касаются исправления обнаруженных ошибок и добавления новой функциональности в систему. Изначально Google рассчитывала давать версиям Android имена известных роботов, но отказалась из-за проблем с авторскими правами. Каждая версия системы, начиная с версии 1.5, получает собственное кодовое имя на тему сладостей. Кодовые имена присваиваются в алфавитном порядке латинского алфавита.

Версии API

API — Расшифровывается как Application Programming Interface (программный интерфейс приложения). Это просто интерфейс, уровень абстракции, который обеспечивает связь между двумя разными «частями» программного обеспечения. Он работает как договор между поставщиком (например, библиотекой) и потребителем (например, приложением). Это набор формальных определений, таких как классы, методы, функции, модули, константы, которые могут использоваться другими разработчиками для написания своего кода. При этом API не включает в себя реализацию.

12	Android 4.1 / 4.2 / 4.3 Jelly Bean
12.1	4.1.1
12.2	4.1.2
12.3	4.2
12.4	4.2.1
12.5	4.2.2
12.6	4.3
12.7	4.3.1
13	Android 4.4 KitKat
13.1	4.4
13.2	4.4.1
13.3	4.4.2
13.4	4.4.3
13.5	4.4.4
14	Android 5.0 / 5.1 Lollipop
14.1	5.0
14.2	5.0.1
14.3	5.0.2
14.4	5.1
14.5	5.1.1
15	Android 6.0 Marshmallow
15.1	6.0
15.2	6.0.1
16	Android 7.0 / 7.1 Nougat
16.1	7.0
16.2	7.1
16.3	7.1.1
16.4	7.1.2
17	Android 8.0 / 8.1 Oreo
17.1	8.0
17.2	8.1
18	Android 9 Pie
18.1	9
19	Android 10
19.1	10
20	Android 11
20.1	11
21	Android 12
21.1	12

Версии API

SDK — Расшифровывается как Software Development Kit (комплект для разработки программного обеспечения). Обратите внимание на слово «kit» (комплект)... он как раз представляет из себя набор различных инструментов, библиотек, документации, примеров, помогающих разработчикам создавать, отлаживать и запускать приложения для Android. API предоставляется вместе с SDK.

Если открыть SDK Manager в Android Studio, можно будет яснее увидеть, из чего состоит Android SDK

В Android между SDK и API существует отношение 1:1, и часто эти два термина используются как синонимы, но важно понимать, что это не одно и то же.

Уровень API — это целочисленное значение, однозначно идентифицирующее версию API фреймворка, предлагаемую платформой Android

12	Android 4.1 / 4.2 / 4.3 Jelly Bean
12.1	4.1.1
12.2	4.1.2
12.3	4.2
12.4	4.2.1
12.5	4.2.2
12.6	4.3
12.7	4.3.1
13	Android 4.4 KitKat
13.1	4.4
13.2	4.4.1
13.3	4.4.2
13.4	4.4.3
13.5	4.4.4
14	Android 5.0 / 5.1 Lollipop
14.1	5.0
14.2	5.0.1
14.3	5.0.2
14.4	5.1
14.5	5.1.1
15	Android 6.0 Marshmallow
15.1	6.0
15.2	6.0.1
16	Android 7.0 / 7.1 Nougat
16.1	7.0
16.2	7.1
16.3	7.1.1
16.4	7.1.2
17	Android 8.0 / 8.1 Oreo
17.1	8.0
17.2	8.1
18	Android 9 Pie
18.1	9
19	Android 10
19.1	10
20	Android 11
20.1	11
21	Android 12
21.1	12

Версии API

compileSdkVersion — Это значение используется только для указания Gradle, с какой версией SDK компилировать ваше приложение. Это позволяет разработчикам получить доступ ко всем API, доступным до уровня API, установленного для compileSdkVersion.

minSdkVersion — Это значение обозначает минимальный уровень API, на котором приложение может работать. Это минимальное требование. Если не указан, значением по умолчанию является 1

Разработчики обязаны установить корректное значение и обеспечить правильную работу приложения до этого уровня API. Это называется обратной совместимостью

targetSdkVersion — Это значение указывает уровень API, на котором приложение было разработано. Является частью APK (также как и minSdkVersion) и изменяет поведение среды выполнения. Это способ, которым разработчики могут контролировать прямую совместимость

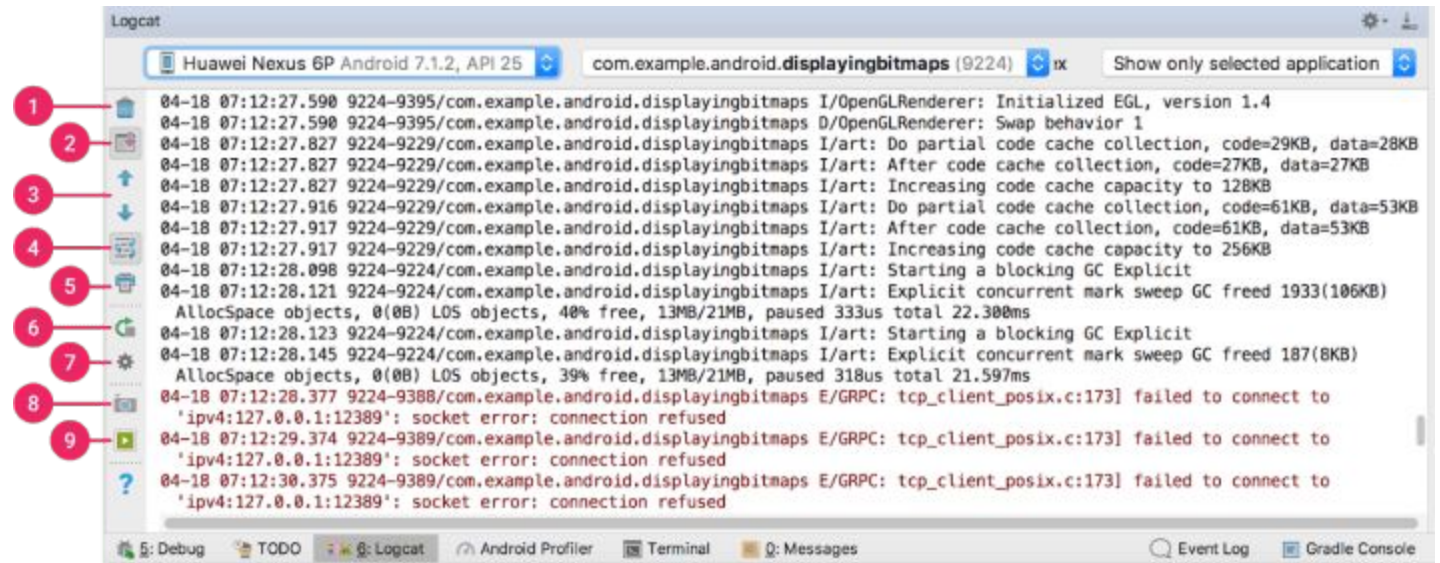
$$\text{minSdkVersion} \leq \text{targetSdkVersion} \leq \text{compileSdkVersion}$$

12	Android 4.1 / 4.2 / 4.3 Jelly Bean
12.1	4.1.1
12.2	4.1.2
12.3	4.2
12.4	4.2.1
12.5	4.2.2
12.6	4.3
12.7	4.3.1
13	Android 4.4 KitKat
13.1	4.4
13.2	4.4.1
13.3	4.4.2
13.4	4.4.3
13.5	4.4.4
14	Android 5.0 / 5.1 Lollipop
14.1	5.0
14.2	5.0.1
14.3	5.0.2
14.4	5.1
14.5	5.1.1
15	Android 6.0 Marshmallow
15.1	6.0
15.2	6.0.1
16	Android 7.0 / 7.1 Nougat
16.1	7.0
16.2	7.1
16.3	7.1.1
16.4	7.1.2
17	Android 8.0 / 8.1 Oreo
17.1	8.0
17.2	8.1
18	Android 9 Pie
18.1	9
19	Android 10
19.1	10
20	Android 11
20.1	11
21	Android 12
21.1	12

Работа с дебагом и логирование

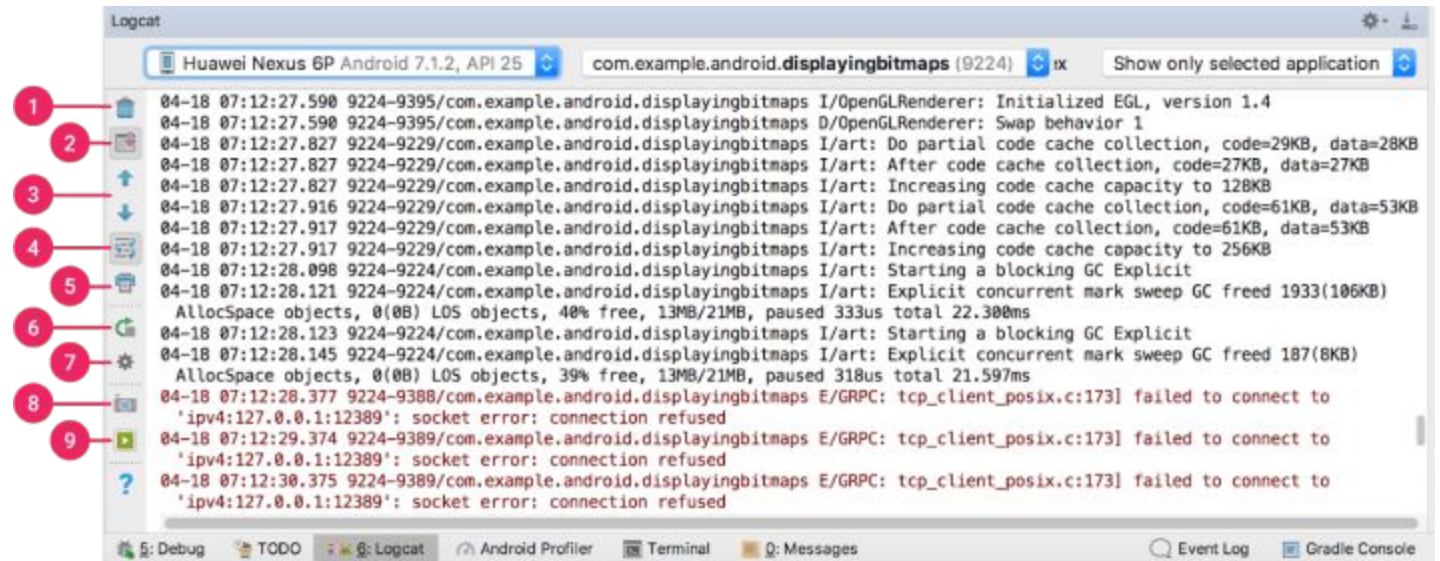
В Android SDK входит набор инструментов, предназначенных для отладки. Самый основной инструмент при отладке - это LogCat. Он отображает сообщения журнала логов, рассылаемые при помощи различных методов

Класс `android.util.Log` позволяет разбивать сообщения по категориям в зависимости от важности. Для разбивки по категориям используются специальные методы, которые легко запомнить по первым буквам, указывающим на категорию



Работа с дебагом и логирование

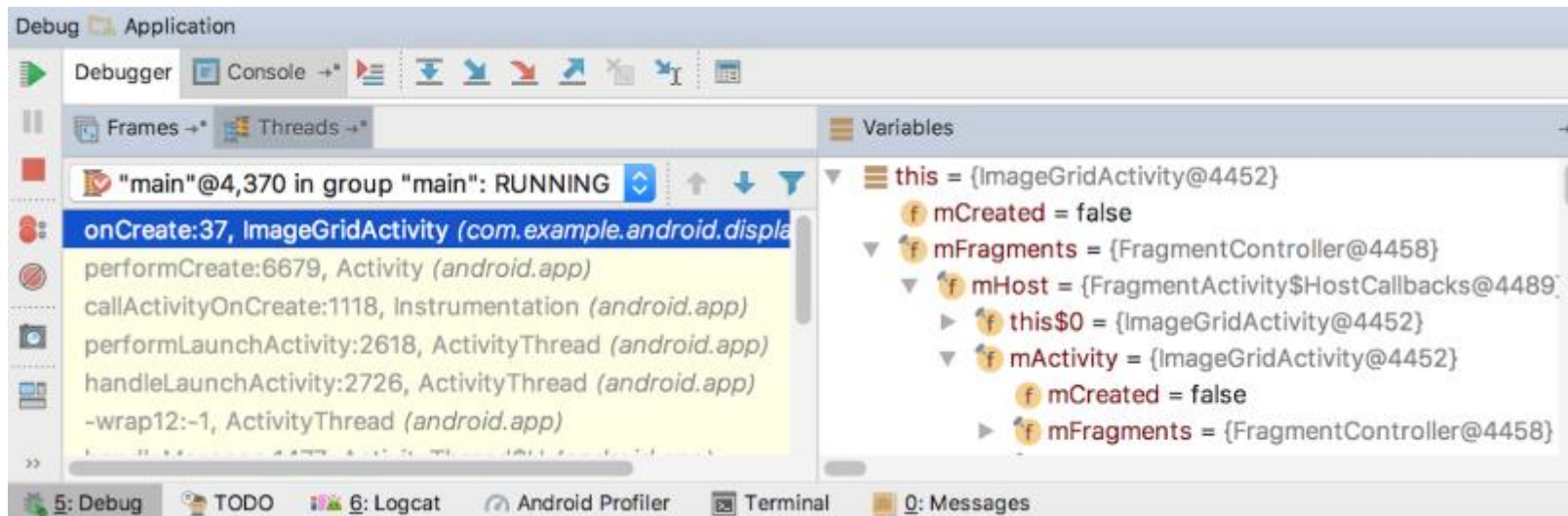
- Log.e() - ошибки (error)
- Log.w() - предупреждения (warning)
- Log.i() - информация (info)
- Log.d() - отладка (debug)
- Log.v() - подробности (verbose)
- Log.wtf() - очень серьезная ошибка!



(What a Terrible Failure!, работает начиная с
Android 2.2)

Работа с дебагом и логирование

```
int id = item.getItemId();  
if (id == R.id.action_settings) {
```



Инструменты командной строки

Инструменты это ничто иное как небольшие консольные утилиты. Располагаются они по пути

android_sdk/

Самые популярные инструмент это:

adb – Отладчик андроид. Поскольку Android это все же Linux, к нему по USB можно подключать другие устройства в том числе смартфоны, adb позволяет ими управлять и дает доступ к их терминалу.

am – Менеджер окон (Action Manager), с помощью него можно запустить и остановить любое приложение или сервис. Например запуск приложения настройки:

```
am start -n com.android.settings/.Settings
```

logcat - просмотр логов приложения

Задания

1. Установить Android Studio. Создать тестовый проект с Empty Activity. Установить имя пакета org.hse.android. Язык Java
2. В MainActivity вывести в лог отладочную информацию, например строку “onCreate”
3. Создать эмулятор для телефона Nexus 4 с API 24
4. Запустить проект и через консольную утилиту logcat вывести и отфильтровать добавленную отладочную информацию в консоль IDE по тэгу или строке
5. Создать точку остановки в методе onCreate и посмотреть значение из отладчика для доступных переменных
6. Собрать тестовый проект в виде установочного файла apk дебаг версии
7. Удалить приложение с эмулятора. Полученный файл установить на эмулятор через консоль с помощью adb

Альтернативные эмуляторы Android

Genymotion (Возможность как локального, так и удаленного запуска)

www.genymotion.com/

ARC Welder (Запуск приложений в Chrome)

chrome.google.com/webstore/detail/arc-welder/emfinbmilocnlhgmfkkmkngdoccbadn

BlueStacks

www.bluestacks.com

Литература

Архитектура мобильных приложений Android :

developer.android.com/topic/libraries/architecture/index.html

Гайд по установке Android Studio:

developer.android.com/studio/intro/index.html

Логирование:

developer.android.com/reference/android/util/Log.html

В целом о пользовательском интерфейсе:

developer.android.com/guide/topics/ui

Инструменты командной строки Android SDK:

developer.android.com/studio/command-line/index.html